

Міністерство освіти і науки України  
Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

## **Інженерія програмного забезпечення**

*Методичні вказівки до виконання лабораторних робіт  
для студентів денної форми навчання  
за спеціальністю 123 “Комп’ютерна інженерія”*

**ЗАТВЕРДЖЕНО**

на засіданні кафедри кібербезпеки та  
програмного забезпечення, протокол № 1  
від 05.07.2017

Кропивницький

2017

Інженерія програмного забезпечення: методичні вказівки до виконання лабораторних робіт для студентів за спеціальністю. 123 «Комп'ютерна інженерія» / М-во освіти і науки України, Центральноукр. нац. техн. ун-т; / уклад. О.А. Смірнов, О.К. Коноплицька-Слободенюк, В.Д. Хох, С.А. Смірнов/ – Кропивницький: ЦНТУ – 2017. – 115 с.

Укладачі: Смірнов О. А., докт. техн. наук, професор;  
Коноплицька-Слободенюк О. К., викладач;  
Хох В.Д., аспірант;  
Смірнов С.А., канд. техн. наук, ст. викладач.

Рецензенти: Сидоренко В. В., докт. техн. наук, професор;  
Доренський О. П., канд. техн. наук.

© Центральноукраїнський  
національний технічний  
університет, 2017

## ЗМІСТ

<b>ВСТУП.....</b>	<b>4</b>
<b>Лабораторна робота №1. Знайомство з MS-Project . Реалізації графіку виконання проекту.....</b>	<b>6</b>
<b>Лабораторна робота №2. Використання ресурсів у проекті .....</b>	<b>36</b>
<b>Лабораторна робота №3. Написання специфікації на програму, що розробляється.....</b>	<b>47</b>
<b>Лабораторна робота №4. Моделювання інформаційних систем з використанням CASE засобів.....</b>	<b>60</b>
<b>Лабораторна робота №5. Розробка програми відповідно специфікації.....</b>	<b>67</b>
<b>Лабораторна робота №6. Створення файлу допомоги до розробленої програми.....</b>	<b>70</b>
<b>Лабораторна робота №7. Організація тестування розробленої програми-додатку.....</b>	<b>72</b>
<b>Лабораторна робота №8. Створення інсталяційного пакету розробленої програми.....</b>	<b>81</b>
<b>Лабораторна робота №9. Створення власного пакета інсталяції програмного забезпечення.....</b>	<b>92</b>
<b>Лабораторна робота №10. Створення власного пакета деінсталяції програмного забезпечення.....</b>	<b>96</b>
<b>Лабораторна робота №11. Використання програм моніторингу жорсткого диска й системного реєстру для перевірки інсталяційних пакетів і програм.....</b>	<b>99</b>
<b>Лабораторна робота №12. Використання файлів конфігурації ОС Windows.....</b>	<b>105</b>
<b>Лабораторна робота №13 Створення програми меню для носія інформації.....</b>	<b>107</b>
<b>Лабораторна робота №14. Створення власного формату файлу .....</b>	<b>110</b>
<b>Список використаної літератури.....</b>	<b>112</b>

## **ВСТУП**

### **Мета викладання дисципліни**

Основна мета курсу полягає в отриманні досконалих знань у області інженерії програмного забезпечення а також отримання студентами навичок створення програмного забезпечення.

Задачі вивчення дисципліни:

- Вивчення теоретичних основ і положень менеджменту програмних проектів.
- Вивчення теоретичних основ і положень вимог до програмного забезпечення.
- Вивчення теоретичних основ і положень моделювання програмного забезпечення.
- Вивчення теоретичних основ і положень стандартів з розробки програмного забезпечення.
- Вивчення теоретичних основ і положень методів забезпечення та контролю якості програмного забезпечення.
- Вивчення теоретичних основ і положень проектування та розробки програмного забезпечення.
- Вивчення теоретичних основ і положень засобів та середовища створення програмного забезпечення.
- Отримання практичних навиків програмування.

Форма підсумкового контролю: іспит.

Максимальну кількість балів студент може одержати у випадку відвідування всіх лекцій, лабораторних занять, виконання і захисту виконаних самостійних завдань у встановлений термін, проходження контролю.

В якості самостійного завдання необхідно виконати теоретичну роботу, згідно обраної студентом теми.

### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
		для екзамену
90 – 100	<b>A</b>	відмінно
82-89	<b>B</b>	добре
74-81	<b>C</b>	
64-73	<b>D</b>	задовільно
60-63	<b>E</b>	
35-59	<b>FX</b>	незадовільно з можливістю повторного складання
0-34	<b>F</b>	незадовільно з обов'язковим повторним вивченням дисципліни

Вибравши предметну область, над якою ви будете працювати, ви повинні виконати завдання до лабораторних робіт, а також відповісти на питання в кінці кожної лабораторної роботи. Звіт повинен містити хід виконання завдань а також графічні матеріали, що підтверджують виконання цих завдань.

## **Лабораторна робота №1 (семестр 5)**

**ТЕМА:** Знайомство з MS-Project . Реалізації графіку виконання проекту.

**МЕТА:** Отримати практичні навички в розробці плану проекту реалізації програмного продукту, чітко ставити цілі перед розробкою програми

**ЗНАТИ:** Середовище MS-Project

**ВМІТИ:** Інсталювати та налаштувати MS-Project.

### **ТЕОРЕТИЧНІ ВІДОМОСТІ.**

Теоретичні відомості з менеджменту проектів та створення проектних груп приведені у лекції № 1.

На перший погляд Microsoft Project здається ще одним додатком сімейства Microsoft Office з панеллю інструментів, як в Word, таблицями й графіками, як в Excel. Але чим глибше ви будете освоювати Microsoft Project, тим більше відмінностей ви побачите.

Одна із ключових відмінностей складається у вузькій області застосування програми. Якщо інші додатки сімейства Microsoft Office орієнтовані на широку область застосування й містять самі різні функції, то програма MS Project призначена винятково для управління проектами.

Інша важлива відмінність полягає в тому, що з MS Project неможливо працювати, не маючи теоретичні знання в області управління проектами й не знаючи особливостей цієї програми. Будь-який користувач може відкрити Word і підготувати документ, не читаючи попередньо книгу. Звичайно, цей користувач не буде застосовувати команди стильового оформлення, можливо, він не зуміє вставити в документ номера сторінок і т.п., але документ буде готовий до друку і їм можна буде скористатися. Якщо ж ви відкриєте MS Project і спробуєте створити план проекту, то без спеціальних знань ви не зможете одержати план, придатний для реалізації проекту.

Таким чином, якщо ви хочете управляти проектами за допомогою MS Project, то без вивчення теорії управління проектами й функціональними можливостями програми вам не обійтися.

Стандартна й професійна редакції MS Project призначені для створення плану проекту, що потім можна опублікувати на сервері MS Project Server для організації спільної роботи над проектом. За допомогою сервера MS Project Server члени проектної команди одержують від керівника завдання, повідомляють про хід їхнього виконання, передоручають їхній один одному. Керівник проекту, відповідно до даних, що надходять від співробітників, відслідковує хід виконання робіт, а керівництво організації аналізує завантаження співробітників і стан всіх проектів, виконуваних в організації.

У якості СУБД, використовуваної MS Project Central, попередником MS Project Server, могли виступати й Oracle, і MS SQL Server. MS Project Server може використовувати в якості СУБД тільки MS SQL Server версій 7 і вище.

Для спільної роботи над проектними документами призначений пакет Share-Point Team Services, що входить у дистрибутив MS Project Server. Цей пакет використовує в якості СУБД MS SQL Server, і його можливості доступні й у стандартної, і в професійній редакції MS Project.

#### *Відмінності стандартної редакції від професійної*

Якщо у вашій організації не використовується сервер MS Project Server, то відмінності між стандартною й професійною редакцією не будуть помітні, оскільки вони пов'язані з можливостями використання сервера MS Project Server.

Комбінація MS Project 2010 стандартної редакції й сервера MS Project Server призначена для спільної роботи невеликих груп над проектами. Комбінація ж професійної редакції MS Project 2010 і сервера призначена для роботи над проектами у великій організації. Тому при використанні професійної редакції сервер дозволяє здійснювати операції, необхідні на великому підприємстві, такі як централізоване зберігання шаблонів проектів і списку співробітників підприємства. Крім того, у професійній версії доступні

засоби автоматизованого підбора співробітників у проект на основі їхніх навичок, а також можливе прогнозування завантаження підприємства з урахуванням певних сценаріїв розвитку подій.

Сервер MS Project Server випускається в одній редакції й може використовуватися й зі стандартної, і із професійною редакцією MS Project 2010, і навіть із MS Project 2000. Виконання професійних операцій на сервері можливо тільки при використанні MS Project 2010 Professional.

*Як вибрати редакцію й що потрібно буде встановлювати*

Якщо ви плануєте працювати з MS Project без використання засобів спільної роботи, то вам потрібно встановити програму в стандартній редакції. Якщо ви плануєте спільну роботу над проектами в рамках невеликої групи, то вам підійде стандартна редакція настільного додатка в сполученні із сервером MS Project Server. При цьому вам не буде потрібно встановлювати СУБД MS SQL Server, тому що в поставку MS Project Server входить його скорочена версія, призначена для використання зі стандартною редакцією.

І нарешті, якщо вам потрібно використовувати корпоративні можливості пакета, то встановлюйте професійну редакцію настільного додатка й сервер MS Project Server. При цьому вам буде потрібно встановити MS SQL Server версії 7 або вище. Для аналітичних операцій, використовуваних у професійній редакції, СУБД повинна підтримувати OLAP, тобто на сервері MS SQL Server повинен бути встановлений пакет Analysis Services.

Далі у першій частині ви познайомитеся з інтерфейсом програми, щоб довідатися можливості інструмента, з яким вам надалі прийдеться працювати. При цьому не потрібно теоретичних знань в області управління проектами: ви будете освоювати принципи роботи з таблицями й діаграмами, учитися вводити, редагувати, сортувати, групувати й фільтрувати дані.

У другій частині ми перейдемо до створення проектного плану. Ви будете створювати власний план проекту й одночасно із практикою планування будете вивчати теорію управління проектами. Ви довідаєтеся про те, як визначати



склад робіт, як розподіляти ресурси, планувати витрати й ризики, як поширювати файл на твердження й порівнювати версії проектів.

### *Основні терміни*

Проект складається із завдань, тобто активностей, спрямованих на досягнення певного результату. Щоб завдання могло бути виконане, на нього виділяються ресурси – матеріальні (устаткування) і робітники (співробітники). Виділення ресурсу на завдання називається призначенням, і в завдання може бути необмежене число призначень.

Завдання можуть поєднуватися в групи (або фази), і завдання, що поєднує інші, називається сумарним. Завершальні завдання, тобто завдання, виконання яких приводить до досягнення важливого проектного результату або завершує фазу, називаються віхами.

Завдання має тривалість, тобто час, необхідне на її виконання. Крім того, завдання характеризується обсягом працевитрат (обсягом робіт) і витратами (або вартістю), необхідними для її виконання.

У плані проекту завдання пов'язані з допомогою залежностей, що визначають порядок виконання завдань відносно один одного. Тривалість проекту складається із проміжків часу від початку самого раннього завдання до закінчення найбільш пізньої з урахуванням залежностей між завданнями. Якщо при збільшенні тривалості завдання збільшується тривалість усього проекту, завдання називається критичним.

Управління проектами полягає в складанні плану й відстеженні виконання робіт з нього. Відповідно, чим краще план проекту, тим більш акуратно він складений, тим легше потім виконувати проектні роботи й вдало завершити проект.

Щоб добре планувати, потрібно, у першу чергу, добре уявляти собі, що таке проект і з яких елементів складається його план.

Діяльність будь-якої організації складається з виконання операцій і проектів. І ті й інші мають багато загального, наприклад виконуються людьми й на їхнє виконання виділяються обмежені ресурси.

Головна відмінність операцій від проектів полягає в тому, що операції йдуть постійно й повторюються, тоді як проекти тимчасові й унікальні. Виходячи із цього, проект визначається як тимчасове зусилля, почате для створення унікального продукту або послуги. «Тимчасове» означає, що кожний проект має точно певні дати початку й закінчення. Говорячи про унікальність продукту або послуги, ми маємо на увазі, що вони мають помітні відмінності від всіх аналогічних продуктів або послуг.

Проекти вживають на будь-яких рівнях організації, і в них можуть бути залучені як кілька людей, так і кілька тисяч. Проекти можуть бути різної тривалості: деякі тривають менш ста годин, інші – більше мільйона. Проект може утягувати один відділ організації, а може й виходити за її межі, як у випадках спільних підприємств і партнерства. Проекти можуть здійснюватися в будь-якій області діяльності. Так, проектами можуть бути й проектування транспортного засобу, і розробка інформаційної системи, і проведення передвиборної кампанії, і будівля будинку, і підготовка номера журналу.

#### *Проект як тимчасове явище*

У кожного проекту є чітко визначені початок і кінець. Кінець проекту настає разом з досягненням всіх його цілей або коли стає ясно, що ці цілі не будуть або не можуть бути досягнуті й проект обривається. Тимчасовість не означає короткостроковість проекту – багато проектів можуть тривати кілька років. У кожному разі, проект кінцевий і не може складатися з постійно триваючих дій.

Дуже багато підприємств тимчасові в тому розумінні, що в якийсь момент робота на них зупиниться. Наприклад, зрозуміло, що конвеєр по виробництву певної моделі автомобілів колись зупиниться, тому що машина буде знята з виробництва. Однак такий рід тимчасовості не робить конвеєр проектом, оскільки робота зі складання машин є типовою рутинною операційною діяльністю. Фундаментальна відмінність проекту полягає в тому, що проект кінчається, коли поставлені цілі досягнуті, тоді як при непроектній діяльності перед виконавцями ставляться нові цілі й робота триває.

Тимчасова природа проектів позначається й на інших аспектах проектної діяльності.

Наприклад, проекти звичайно мають дуже обмежені тимчасові рамки для створення продукту або послуги, оскільки сприятлива для них ситуація на ринку складається на обмежений час. Крім того, проектна команда, як правило, по його закінченні розпадається, а її члени переходять в інші проекти.

На відміну від конвеєра по складанню автомобілів, гарним прикладом проекту може бути розробка нового автомобіля. Розробка здійснюється в обмежені тимчасові строки й для досягнення певного результату – прототипу нового автомобіля. Коли результат досягнуть, автомобіль відправляється у виробництво, а проектна команда – конструктори, дизайнери, інженери та ін. можуть бути залучені в новий проект, хоча й не обов'язково в тому же складі. Проект дуже часто плутають із програмою, тобто координованим управлінням групою проектів усередині однієї організації. Управління відразу декількома проектами скоординоване для того, щоб одержати вигоду, яку не можна одержати від окремого управління кожним з них. Програми звичайно сполучають елементи проектів і операцій. Наприклад, розробка веб-сайту є проектом, тоді як підтримка його протягом тривалого часу – це операційна діяльність.

Програми можуть також включати повторювані або циклічні роботи, наприклад видання журналу: періодичне видання саме по собі є безперервним процесом, тоді як підготовка окремого номера – це проект.

Проект вживається для досягнення певного результату в певний термін і за певні гроші. План проекту складається для того, щоб визначити, за допомогою яких робіт буде досягатися результат проекту, які люди й устаткування потрібні для виконання цих робіт і в який час ці люди й устаткування будуть зайняті роботою із проекту. Тому проектний план містить три основних елементи: завдання (Task), ресурси (Resource) і призначення (Assignment). Розглянемо докладніше кожний з них.

Завдання

Завданням називається робота, здійснювана в рамках проекту для досягнення певного результату. Наприклад, у проекті видання номера журналу завданням буде Проведення редколегії. Оскільки звичайно проект містить багато завдань, то для зручності відстеження плану їх поєднують у групи, або фази. Сукупність фаз проекту називається його життєвим циклом.

### *Фази*

Фаза проекту складається з одного або декількох завдань, у результаті виконання яких досягається один або кілька основних результатів проекту. Таким чином, результати, досягнуті завдяки виконанню кожної із завдань, що входять у фазу, формують її результат.

Якщо для досягнення результатів завдання потрібно виконати тільки його, то для досягнення результату фази потрібно виконати групу інших завдань. І в цьому полягає відмінність фази від завдання: її результат підсумує результати інших завдань. Саме тому в MS Project фази називаються Summary task (Сумарне завдання).

Наприклад, результатом фази Підготовка матеріалів будуть матеріали номера журналу, які можна передати на переддрукову підготовку. Оскільки номер складається з обкладинки й статей, то для одержання результату фази потрібно здійснити як мінімум два завдання: Підготовка обкладинки, результатом якого буде обкладинка журналу з фотографією фотомоделі й заголовками статей, і Підготовка статей, результатом якої будуть всі тексти статей журналу.

Фази можуть складатися як із завдань, так і з інших фаз. Наприклад, Підготовка обкладинки теж є фазою, оскільки може бути розділена на три завдання: Відбір моделі, результатом якого буде прізвище моделі для фотозйомки, Фотозйомка моделі, що закінчується одержанням фотографії відібраної моделі, і Верстка обкладинки. По завершенні останнього завдання зроблена фотографія буде розміщена на обкладинці й обкладинка буде підготовлена до публікації. Проект розбивається на фази й для зручності контролювання ходу роботи. По завершенні проектної фази звичайно

здійснюється аналіз як отриманих результатів, щоб з мінімальними витратами визначити й виправити помилки, так і загального ходу проекту, щоб визначити, чи варто переходити до виконання наступної фази проекту.

Використання перетинання фаз при плануванні називається швидким шляхом (fast tracking).

Розбивка проекту на фази дозволяє представити його у вигляді списку основних результатів і дат, до яких вони повинні бути отримані. Керівник проекту здійснює безпосередній контроль виконання кожного завдання усередині проекту, повідомляючи вищестоящого менеджера тільки про досягнення фазових результатів. Цьому менеджеріві, у свою чергу, для контролю виконання проекту цілком достатньо таких даних.

#### *Завершальні завдання*

Кожний проект вживається для досягнення певної мети, і звичайно досягти її не можна, не досягши декількох проміжних цілей. Наприклад, не можна побудувати будинок, не заклавши фундамент. Закладка фундаменту буде проміжною метою при будівлі будинку.

Завдання, у результаті виконання яких досягаються проміжні цілі, називаються завершальними завданнями. В MS Project вони називаються віхами (Milestone). Звичайно результатом фази є досягнення проміжної мети, тому віхою в плані проекту прийнято позначати останнє завдання фази, у результаті якої досягається її результат.

Іноді, якщо такого завдання немає, а фазовий результат досягається, наприклад одночасним завершенням декількох завдань, то створюється фіктивне завершальне завдання. Тривалість такого завдання встановлюється в 0 днів, і на неї не виділяються виконавці. Вона присутня в плані винятково для позначення моменту завершення фази, що полегшує відстеження плану проекту.

#### *Тривалість і працевитрати*

Тривалість завдання – це період робочого часу, що необхідний для того, щоб виконати його.

УВАГА. При підрахунку тривалості завдання MS Project не враховує неробочий час, наприклад вихідні дні, і перерви в її виконанні.

Тривалість може не відповідати працевитратам співробітника, що займається завданням. Наприклад, для виконання завдання Збір пропозицій від авторів співробітникові (відповідальному секретареві журналу) потрібно витратити півгодини на розсилання типового електронного листа авторам і півгодини на обробку відповідей, що надійшли в плин тижня. Виходить, його працевитрати на виконання завдання складуть одну годину. Таким чином, тривалість відповідає часу, через яке буде отриманий результат завдання, а працевитрати – часу, витраченому співробітниками на одержання результату. В MS Project тривалість позначається терміном Duration, а працевитрати – терміном Work.

#### *Залежності й зв'язки*

Завдання в плані проекту взаємозалежне, наприклад, часто одне завдання не може початися, поки не закінчене інше (зведення стін не може початися раніше закладки фундаменту). В MS Project залежності називаються терміном Dependencies.

На плані проекту залежності позначаються за допомогою зв'язків (Link), і обоє ці терміни – залежність і зв'язок – використовуються з тим самим змістом, позначаючи логіку, що визначає послідовність робіт у плані проекту.

#### *Ролі й ресурси*

Під ресурсами в MS Project розуміються співробітники й устаткування, необхідні для виконання проектних завдань. Наприклад, для виконання завдання Збір пропозицій від авторів у проекті повинен бути задіяний відповідальний секретар журналу.

Кожний співробітник, що бере участь у проекті, одержує певну роль у відповідності зі своєю кваліфікацією, вимогами проекту й регламентами, що діють в організації. Наприклад, в одному проекті співробітник може виступати в ролі архітектора додатків, а в іншому, де гостро потрібен програміст, той же співробітник може бути задіяний у ролі програміста.

При складанні списку ресурсів часто використовується рольове планування. Наприклад, спочатку визначається, що для виконання робіт потрібні три програмісти й один менеджер, а потім, коли план проекту затверджений, вибираються конкретні співробітники для участі в цих ролях.

### *Вартість ресурсів*

Важлива властивість ресурсів – вартість (Cost (Витрати)) їхнього використання в проекті. В MS Project є два типи вартості ресурсів: погодинна ставка й вартість за використання. Погодинна ставка (Rate) виражається у вартості використання ресурсу в одиницю часу, наприклад 100 грн. у годину або 1000 грн. у день. У такому випадку вартість участі ресурсу в проекті складе час, протягом якого він працює в проекті, помножене на погодинну ставку. Звичайно погодинна ставка використовується для обліку вартості нематеріальних ресурсів.

Термін «вартість» (cost) в MS Project переведений як «витрати», тому що в управлінні проектами використовуються обоє українськомовних терміна. Величина Cost Per Use (Витрати на використання) позначає вартість використання встаткування або співробітника в завданні, що не залежить від того, скільки часу задіяне в завданні співробітник або матеріальний ресурс. Загальні витрати на використання ресурсу визначаються шляхом множення вартості використання на число завдань, у яких він задіяний.

У ресурсу може бути зазначена вартість як одного із двох типів, так і обох. При визначенні загальних витрат на використання ресурсу в проекті MS Project визначає погодинні витрати й витрати на використання й підсумує їх.

### *Призначення*

Призначення – це зв'язок певного завдання й ресурсів, необхідних для його виконання. При цьому на одне завдання можуть бути призначені кілька ресурсів, як матеріальних, так і нематеріальних.

Призначення поєднують у плані ресурси й завдання, роблячи план цілісним. Завдяки призначенням вирішується цілий ряд завдань планування. По-перше, визначаються відповідальні за виконання завдань. По-друге, коли

визначені завдання, за які відповідає ресурс, можна розрахувати загальний обсяг часу, затрачуваний їм на проект, а виходить, його вартість для проекту. По-третє, визначивши вартість участі всіх ресурсів у проекті, можна підрахувати його загальну вартість. Нарешті, призначаючи ресурси на завдання, можна скорочувати строк виконання робіт, виділяючи на них більше ресурсів і тим самим скорочуючи загальну тривалість проекту.

Складання плану проекту в загальному виді полягає в описі завдань проекту, доступних ресурсів і визначенні взаємозв'язків між ними за допомогою призначень. Але при складанні плану проекту в MS Project кількість операцій трохи збільшується.

Планування починається з визначення проекту, тобто опису його ключових характеристик. Потім складається список фаз і завдань і список необхідних для їхнього виконання ресурсів. Після цього в план вноситься додаткова інформація про завдання й ресурси, що буде використовуватися при визначенні призначень і надалі при проведенні робіт із плану (відстеженні плану). Нарешті, здійснюються призначення, після чого проект оптимізується, якщо тривалість або бюджет виявляються більше очікуваних.

## **ХІД ВИКОНАННЯ ЛАБОЛАТОРНОЇ РОБОТИ**

### **1. Встановити та налаштувати середовище MS-Project.**

Для цього виконуємо наступні дії.

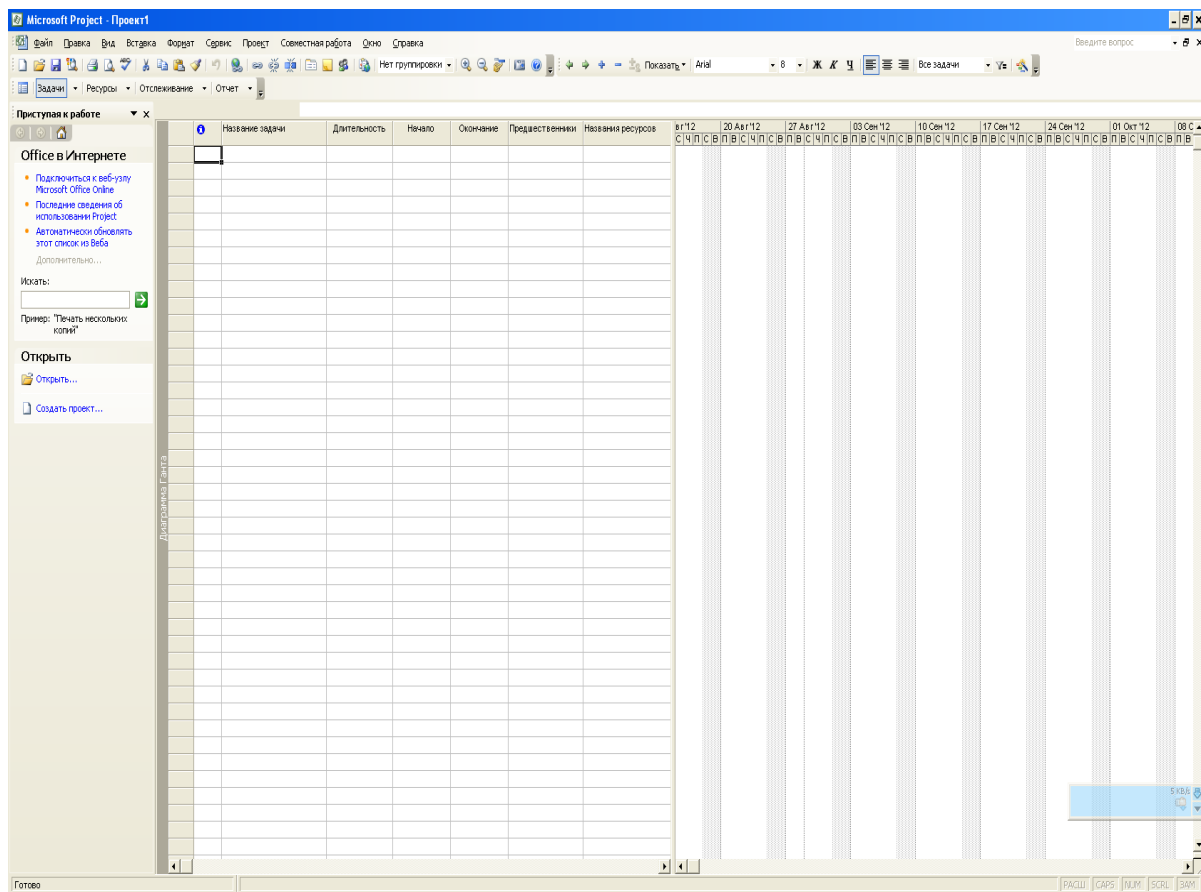
– Встановлюємо з інсталяційного пакету MS-Project.

– Запускаємо встановлений MS-Project наступним чином:

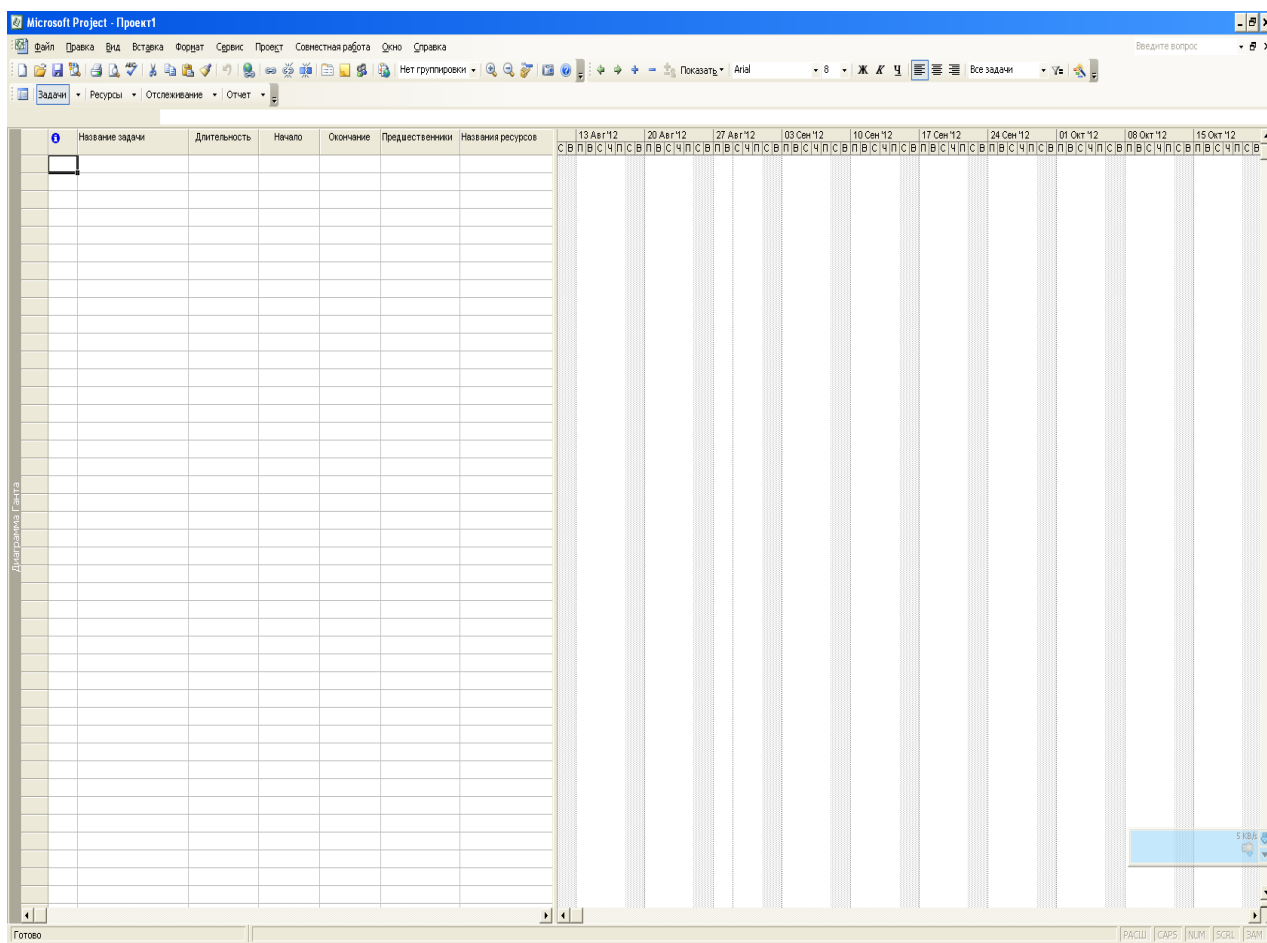
« Пуск → Программы → Microsoft Office → Microsoft Office Project»



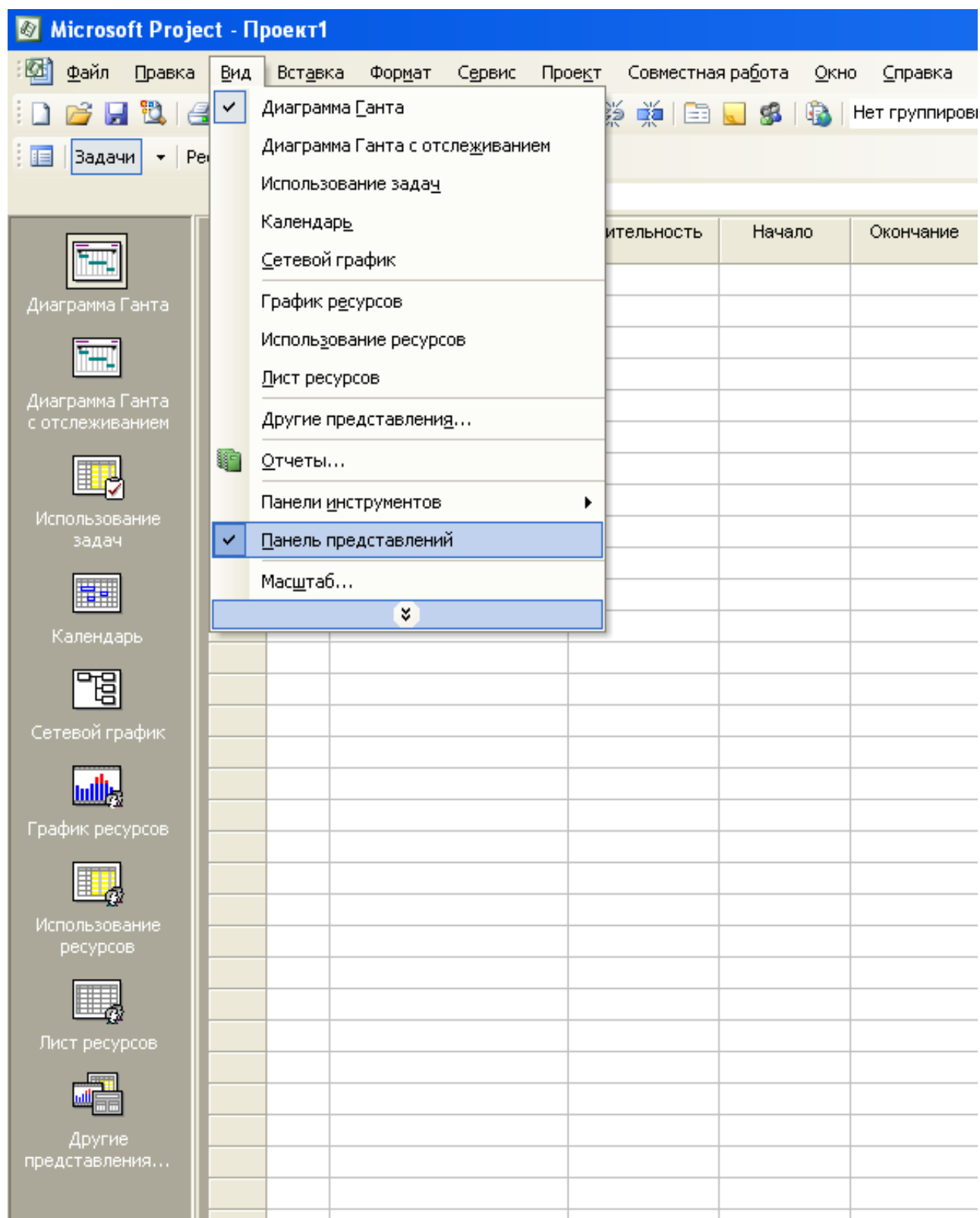
На екрані бачимо наступний вигляд інтерфейсу програми:



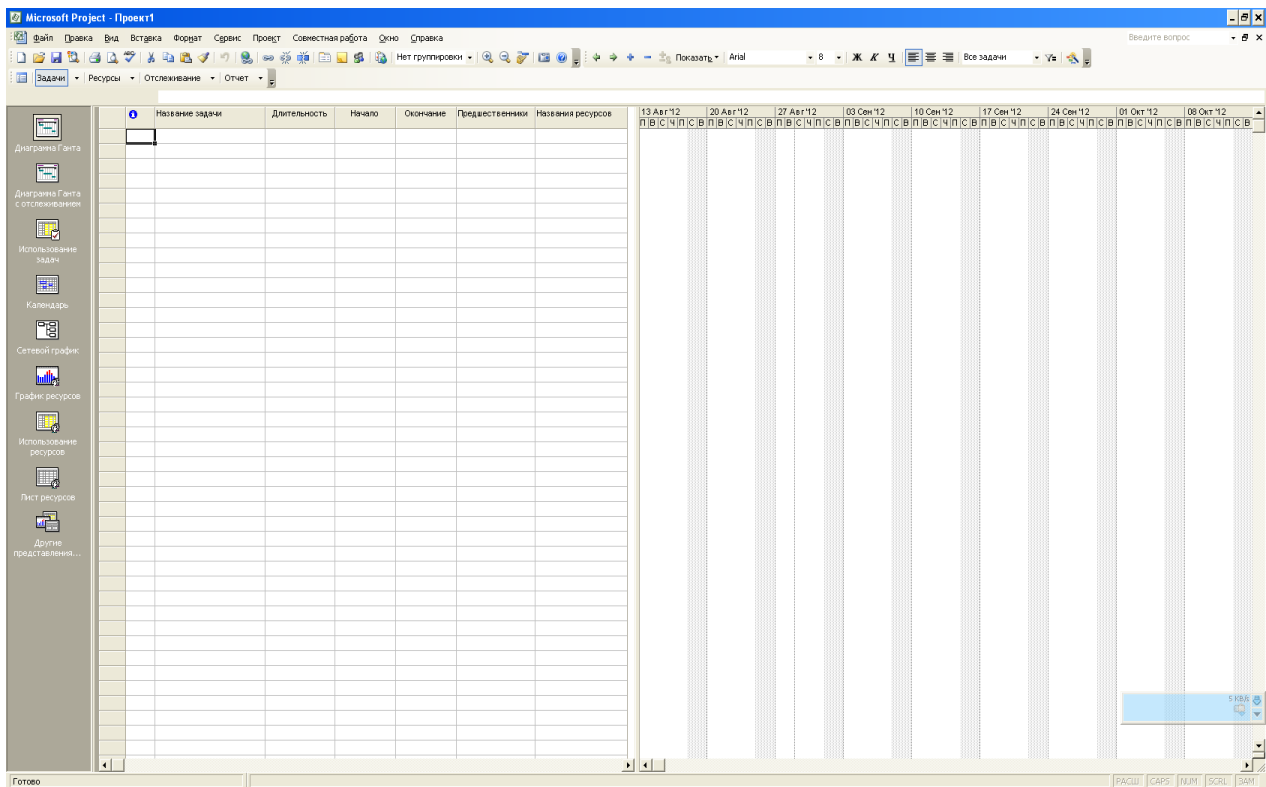
Закриваємо непотрібні вікна з лівого краю:



Налаштовуємо панель представлень (Вид → Панель представлень):



Отримуємо наступний вигляд інтерфейсу:



Зліва знаходиться панель представлень, яка складається з наступних панелей:

Діаграма Ганта.

Діаграма Ганта з відстеженням.

Використання задач.

Календар.

Мережний графік.

Використання ресурсів.

Лист ресурсів.

Інші представлення.

На рисунку діаграма Ганта зображена у правій половині рисунка. Це є графічне відображення календарного виконання проекту.

У лівій частині рисунка знаходиться поле у якому задаються завдання, при цьому є можливість задати наступні параметри:

Назву завдання.

Тривалість.

Початок.

Закінчення.

Попередники.

Назву ресурсів.

2. Починаємо реалізовувати план проекту.

Наведемо приклад: Необхідно реалізувати проект з написання програмного продукту.

Він складається з наступних задач:

- Отримання завдання на реалізацію програмного продукту.
- Розробка плану та концепції реалізації програмного продукту.
- Написання програми.
- Тестування.
- Впровадження.
- Підтримка програмного продукту.

Кожна з задач має свої підзадачі, й тоді проект буде складатися наступним чином:

1. Отримання завдання на реалізацію програмного продукту:

- Зустріч з замовником.
- Підписання договору.
- Отримання завдання.

2. Розробка концепції та плану реалізації програмного продукту:

- Розробка концепції програмного продукту.
- Затвердження концепції.
- Розробка специфікації.
- Розробка календарного плану.
- Узгодження з замовником та затвердження специфікації й календарного плану.

3. Написання програми:

- Вибір мови програмування.
- Написання програмного продукту.

#### 4. Тестування:

- Вибір методів та систем тестування.
- Тестування програмного продукту.
- Обробка результатів тестування.
- Усунення помилок.

#### 5. Впровадження:

- Написання файлу допомоги.
- Встановлення програмного продукту.
- Навчання персоналу користуванню програмний продуктом.

#### 6. Підтримка програмного продукту:

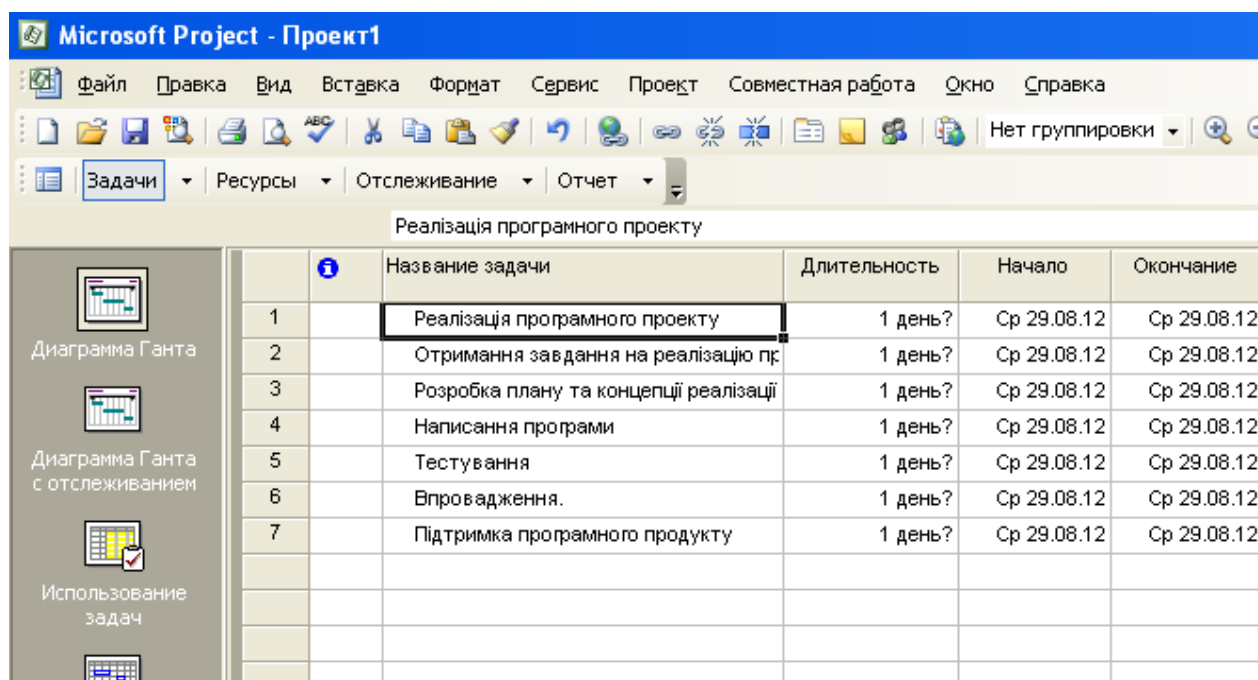
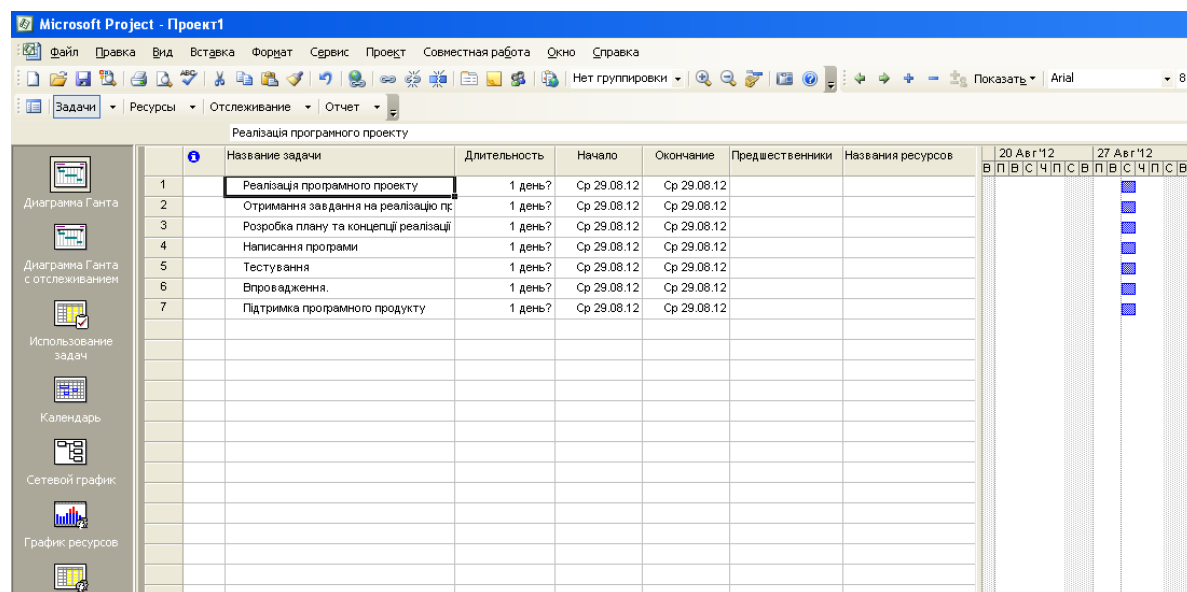
- Реалізація техпідтримки.
- Випуск оновлень.
- Випуск нових версій продукту.

Розглянемо реалізацію цього проекту.

У поле назва завдань, впишемо назву проекту та назви усіх завдань.

У це поле вписуються тільки назви завдань, тобто те, що треба зробити, а не те, хто це буде робити!!!! Ті, хто будуть це робити – це ресурси!!!

## Проект назвемо «Реалізація програмного проекту»

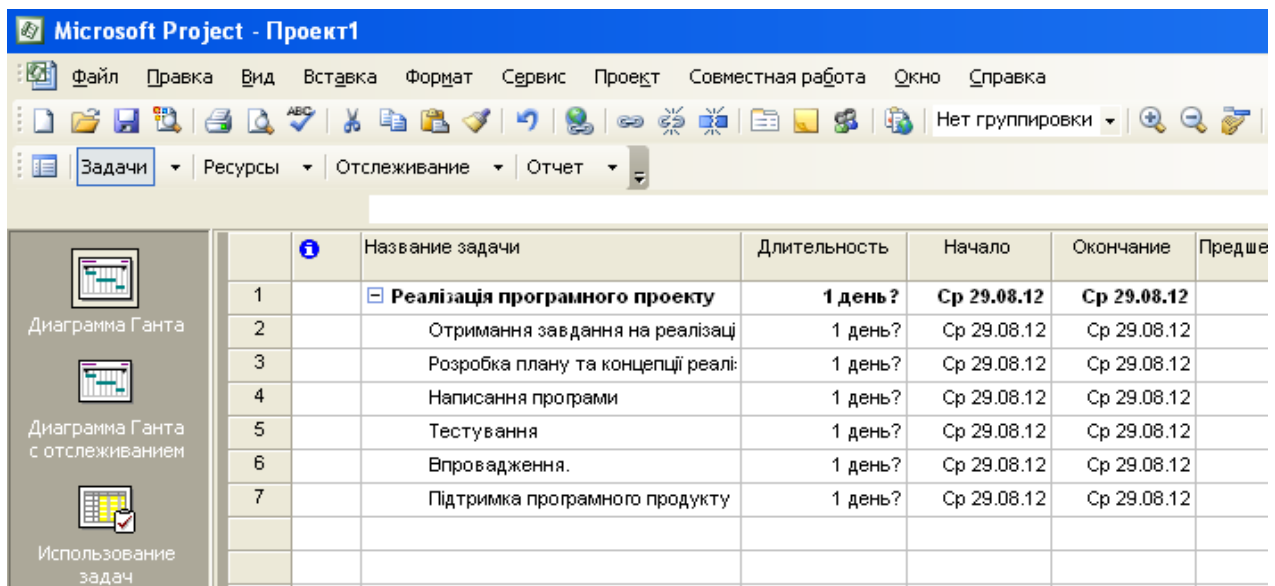


Усі задачі, які знаходяться нижче назви проекту, є для нього підзадачами, тому робимо їх підзадачами.

Для цього, лівою кнопкою миші, виділяємо усі задачі, крім назви проекту.

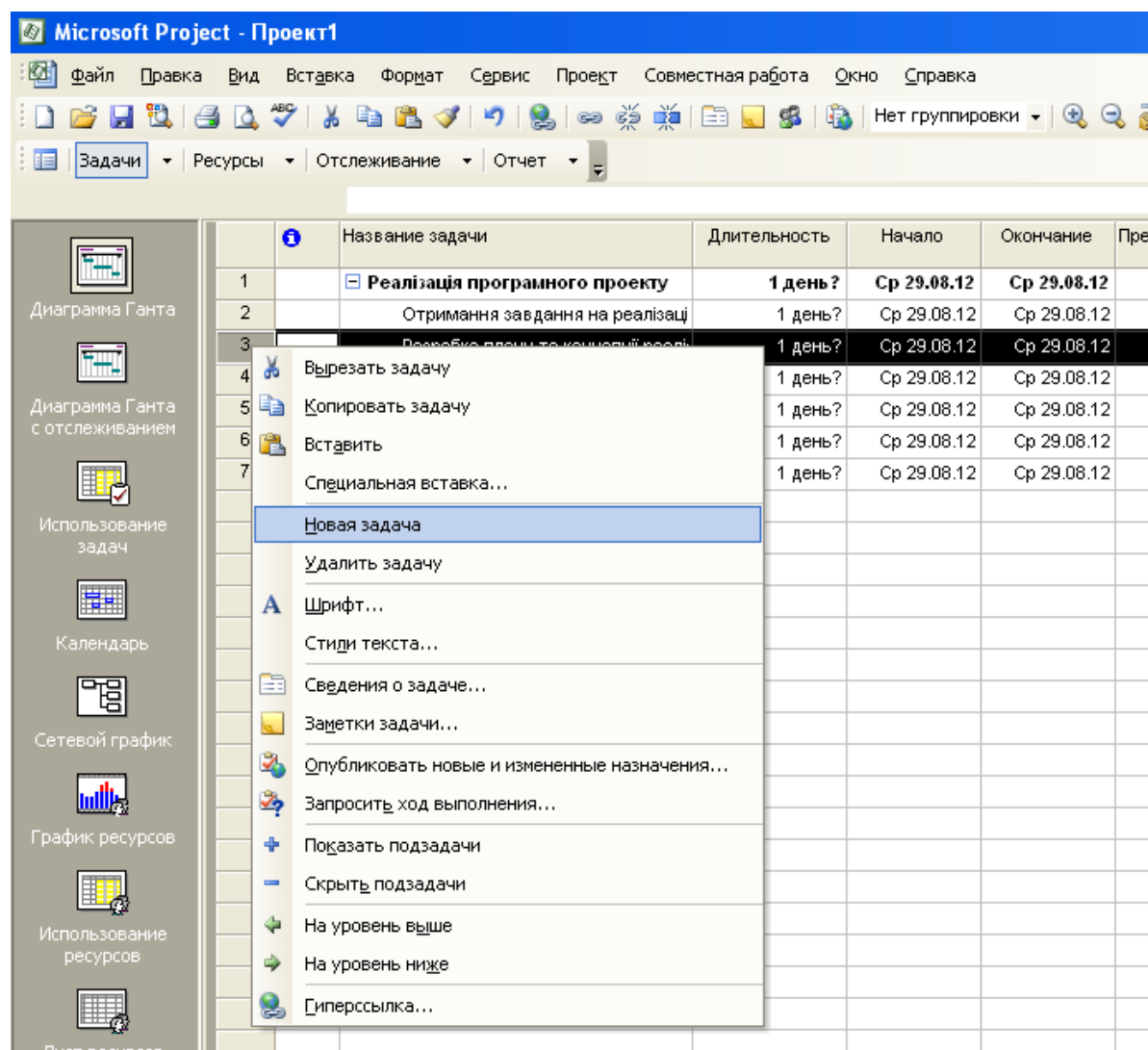






Наступним кроком додамо підзадачі для кожного з завдань.

Для цього наведемо курсор на задачу, яка знаходиться нижче й натиснемо праву кнопку миші. У контекстному меню, яке з'явиться виберемо пункт «Нова задача».



Після виконання цієї дії поле прийме наступний вигляд.

Microsoft Project - Проект1						
<div> <div>Файл</div> <div>Правка</div> <div>Вид</div> <div>Вставка</div> <div>Формат</div> <div>Сервис</div> <div>Проект</div> <div>Совместная работа</div> <div>Окно</div> <div>Справка</div> </div> <div> <div>Нет группировки</div> </div> <div> <div>Задачи</div> <div>Ресурсы</div> <div>Отслеживание</div> <div>Отчет</div> </div>						
	И	Название задачи	Длительность	Начало	Окончание	Предшествующие
1		Реалізація програмного проекту	1 день?	Ср 29.08.12	Ср 29.08.12	
2		Отримання завдання на реалізаці	1 день?	Ср 29.08.12	Ср 29.08.12	
3						
4		Розробка плану та концепції реалі	1 день?	Ср 29.08.12	Ср 29.08.12	
5		Написання програми	1 день?	Ср 29.08.12	Ср 29.08.12	
6		Тестування	1 день?	Ср 29.08.12	Ср 29.08.12	
7		Впровадження	1 день?	Ср 29.08.12	Ср 29.08.12	
8		Підтримка програмного продукту	1 день?	Ср 29.08.12	Ср 29.08.12	

У нашому випадку, для першої задачі існує 3 підзадачі, тому створюємо 3 нові задачі, за таким же принципом, який описаний вище.

Microsoft Project - Проект1

Файл Правка Вид Вставка Формат Сервис Проект Совместная работа Окно Справка

Задачи Ресурсы Отслеживание Отчет

	1	Название задачи	Длительность	Начало	Окончание	Предшествен
1		Реалізація програмного проекту	1 день?	Ср 29.08.12	Ср 29.08.12	
2		Отримання завдання на реалізаці	1 день?	Ср 29.08.12	Ср 29.08.12	
3						
4						
5						
6		Розробка плану та концепції реалі:	1 день?	Ср 29.08.12	Ср 29.08.12	
7		Написання програми	1 день?	Ср 29.08.12	Ср 29.08.12	
8		Тестування	1 день?	Ср 29.08.12	Ср 29.08.12	
9		Впровадження.	1 день?	Ср 29.08.12	Ср 29.08.12	
10		Підтримка програмного продукту	1 день?	Ср 29.08.12	Ср 29.08.12	

Диаграмма Ганта  
Диаграмма Ганта с отслеживанием  
Использование задач  
Календарь

У ці 3 поля вписуємо підзадачі для першої задачі.

Microsoft Project - Проект1

Файл Правка Вид Вставка Формат Сервис Проект Совместная работа Окно Справка

Задачи Ресурсы Отслеживание Отчет

Отслеживание

	1	Название задачи	Длительность	Начало	Окончание	Предшественники	Название ресурсов
1		Реалізація програмного проекту	1 день?	Ср 29.08.12	Ср 29.08.12		
2		Отримання завдання на реалізаці	1 день?	Ср 29.08.12	Ср 29.08.12		
3		Зустріч з замовником	1 день?	Ср 29.08.12	Ср 29.08.12		
4		Підписання договору	1 день?	Ср 29.08.12	Ср 29.08.12		
5		Отримання завдання	1 день?	Ср 29.08.12	Ср 29.08.12		
6		Розробка плану та концепції реалі:	1 день?	Ср 29.08.12	Ср 29.08.12		
7		Написання програми	1 день?	Ср 29.08.12	Ср 29.08.12		
8		Тестування	1 день?	Ср 29.08.12	Ср 29.08.12		
9		Впровадження.	1 день?	Ср 29.08.12	Ср 29.08.12		
10		Підтримка програмного продукту	1 день?	Ср 29.08.12	Ср 29.08.12		

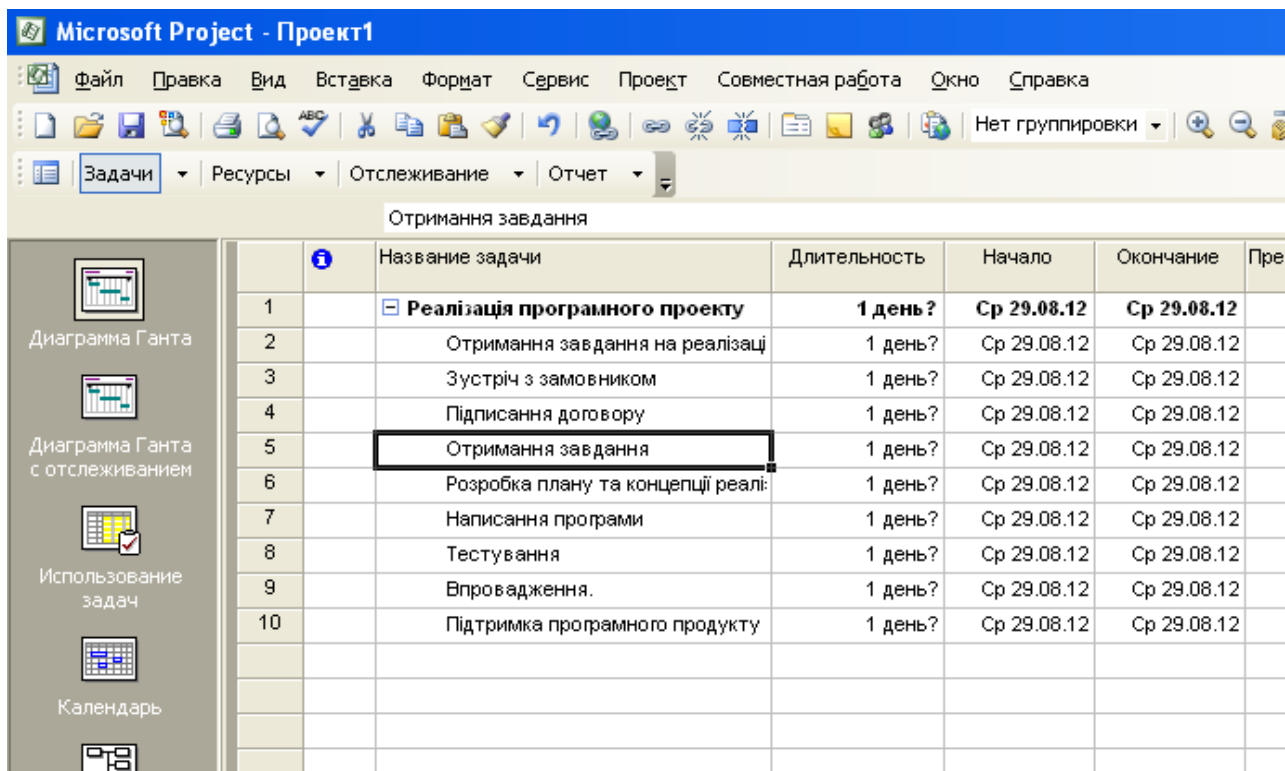
Диаграмма Ганта  
Диаграмма Ганта с отслеживанием  
Использование задач  
Календарь  
Сетевой график  
График ресурсов  
Использование ресурсов  
Лист ресурсов  
Другие представления...


20 Авг '12 27 Авг '12 03 Сен '12 10 Сен '12 17 Сен '12 24 Сен '12 01 Окт '12 08 Окт '12 15

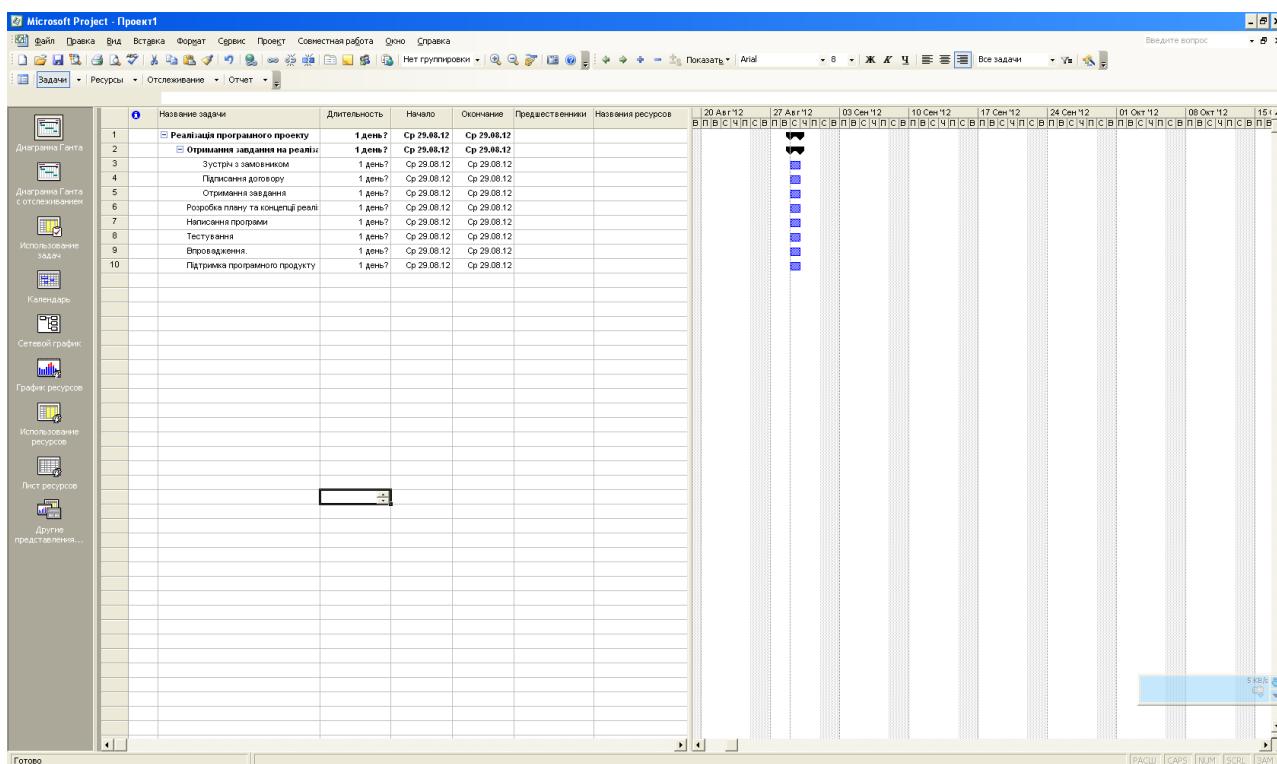
Введите вопрос

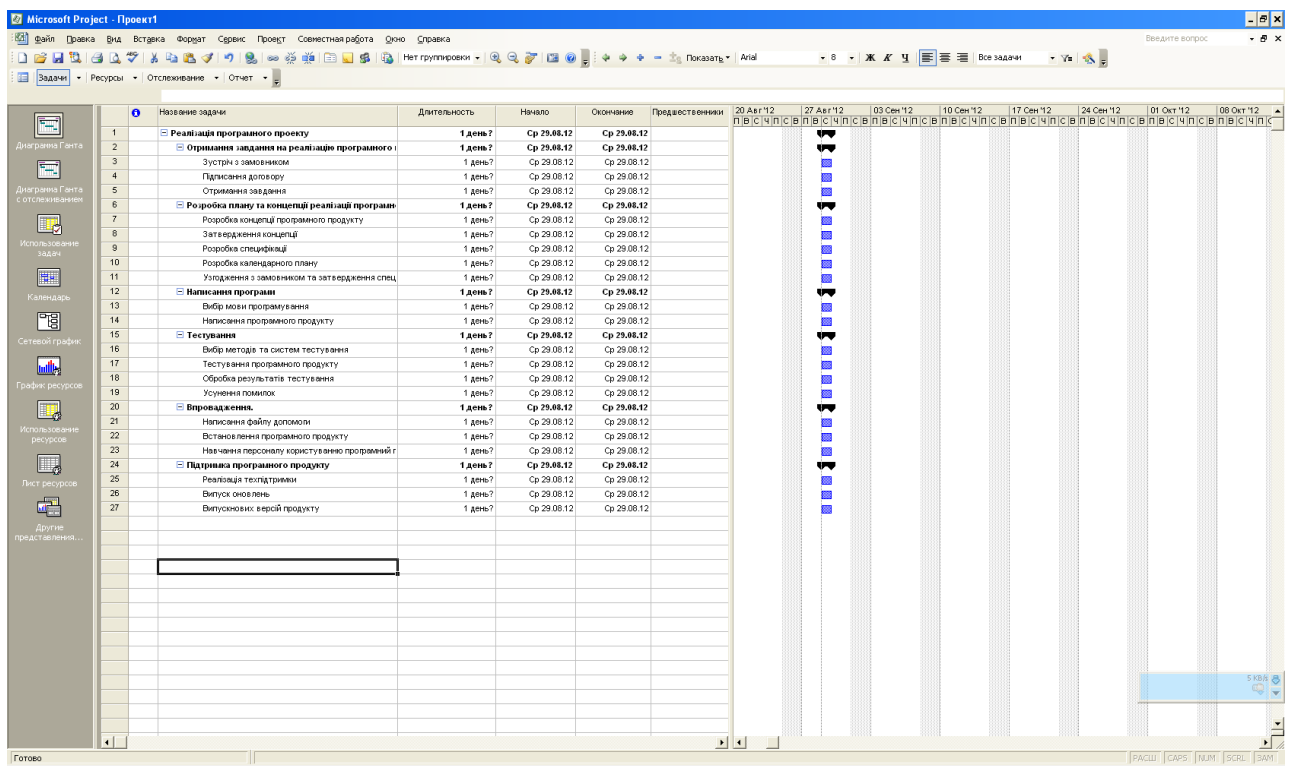
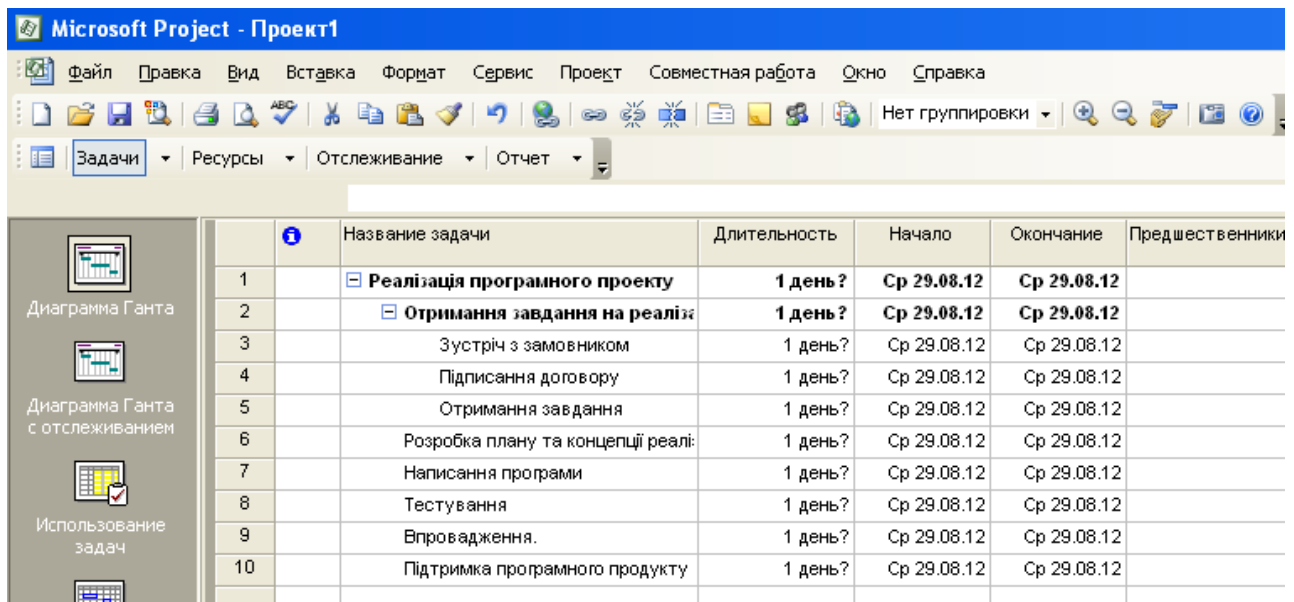
Готово

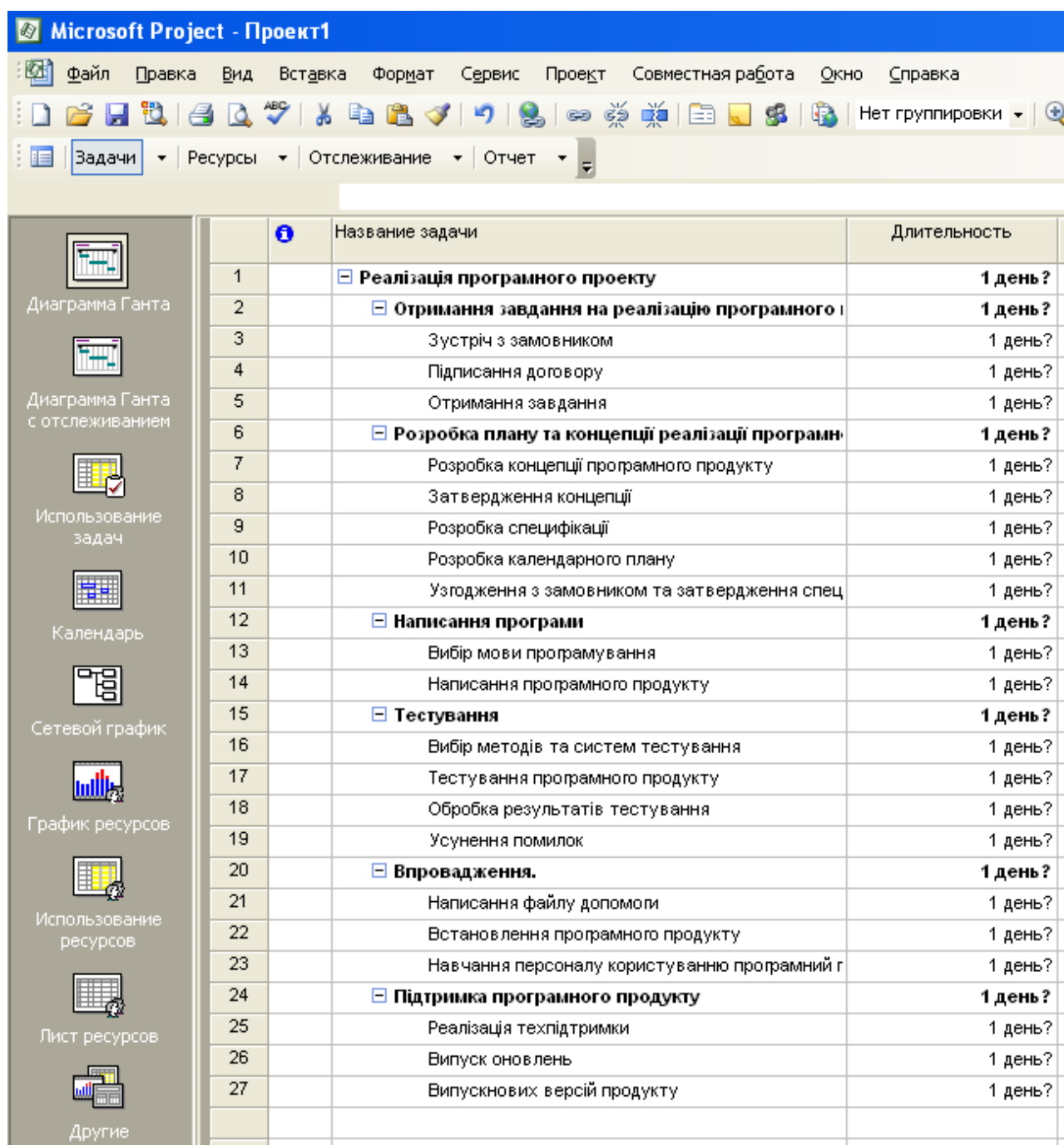
ПACU CAPS NUM SCRL SAM



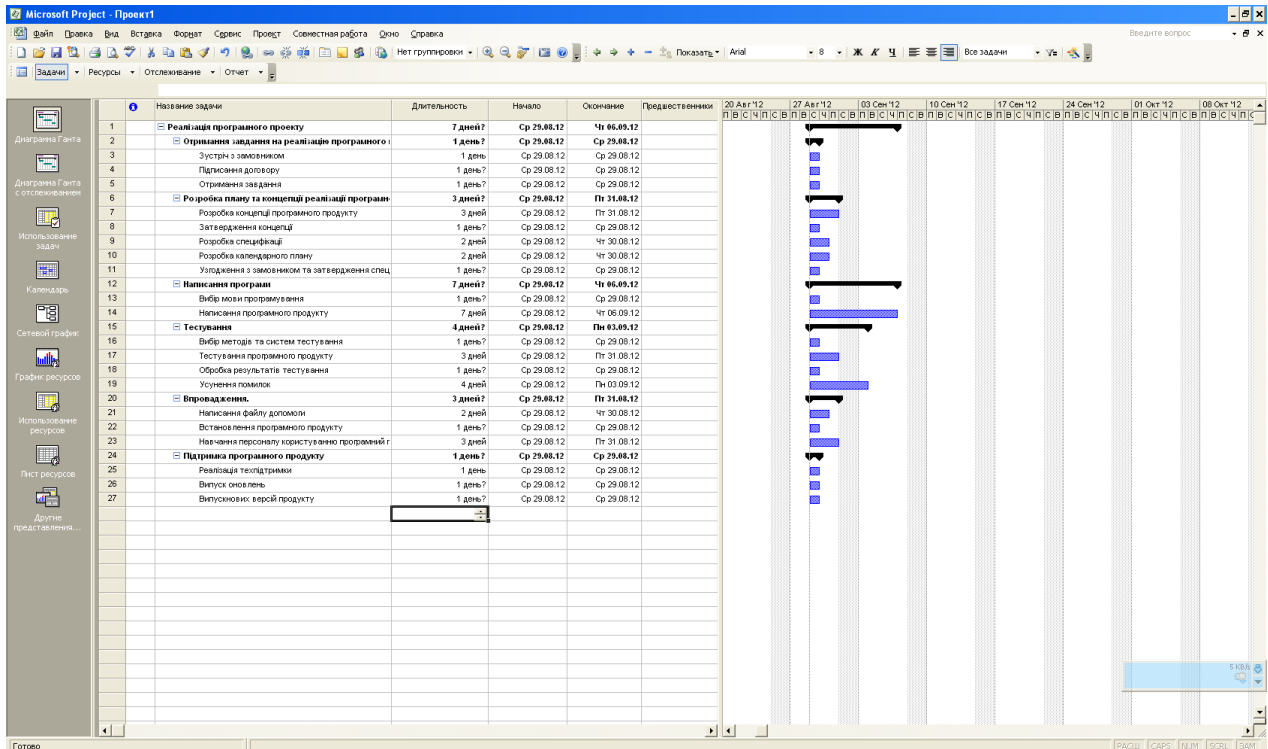
І за алгоритмом наведеним вище, за допомогою кнопки , переводимо їх у стан підзадач.







У полі «Тривалість» виставимо кількість днів для кожної задачі.

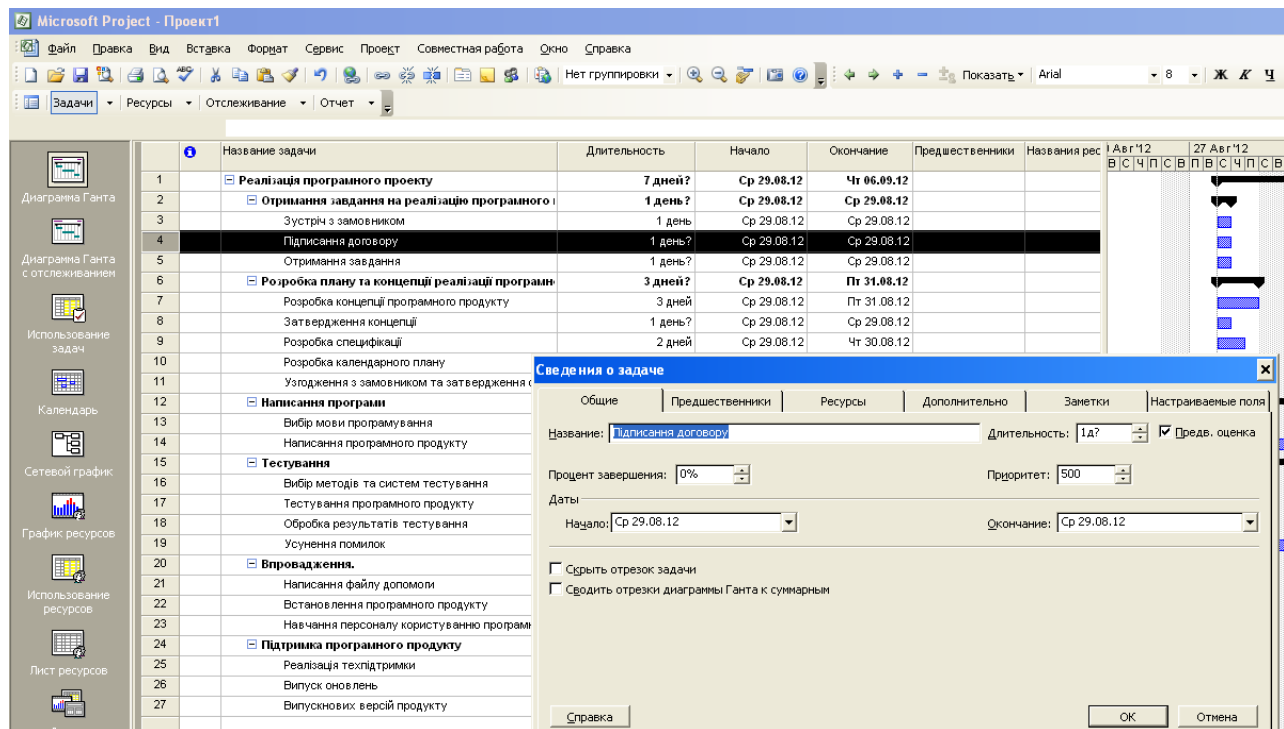


Microsoft Project - Проект1						
Файл Правка Вид Вставка Формат Сервис Проект Совместная работа Окно Справка						
Задачи Ресурсы Отслеживание Отчет						
	№	Название задачи	Длительность	Начало	Окончание	Пред
	1	Реализация программного проекта	7 дней?	Ср 29.08.12	Чт 06.09.12	
	2	Отримання завдання на реалізацію програмного і	1 день?	Ср 29.08.12	Ср 29.08.12	
	3	Зустріч з замовником	1 день?	Ср 29.08.12	Ср 29.08.12	
	4	Підписання договору	1 день?	Ср 29.08.12	Ср 29.08.12	
	5	Отримання завдання	1 день?	Ср 29.08.12	Ср 29.08.12	
	6	Розробка плану та концепції реалізації програмн	3 дней?	Ср 29.08.12	Пт 31.08.12	
	7	Розробка концепції програмного продукту	3 дней?	Ср 29.08.12	Пт 31.08.12	
	8	Затвердження концепції	1 день?	Ср 29.08.12	Ср 29.08.12	
	9	Розробка специфікації	2 дней?	Ср 29.08.12	Чт 30.08.12	
	10	Розробка календарного плану	2 дней?	Ср 29.08.12	Чт 30.08.12	
	11	Узгодження з замовником та затвердження спец	1 день?	Ср 29.08.12	Ср 29.08.12	
	12	Написання програми	7 дней?	Ср 29.08.12	Чт 06.09.12	
	13	Вибір мови програмування	1 день?	Ср 29.08.12	Ср 29.08.12	
	14	Написання програмного продукту	7 дней?	Ср 29.08.12	Чт 06.09.12	
	15	Тестування	4 дней?	Ср 29.08.12	Пн 03.09.12	
	16	Вибір методів та систем тестування	1 день?	Ср 29.08.12	Ср 29.08.12	
	17	Тестування програмного продукту	3 дней?	Ср 29.08.12	Пт 31.08.12	
	18	Обробка результатів тестування	1 день?	Ср 29.08.12	Ср 29.08.12	
	19	Усунення помилок	4 дней?	Ср 29.08.12	Пн 03.09.12	
	20	Впровадження.	3 дней?	Ср 29.08.12	Пт 31.08.12	
	21	Написання файлу допомоги	2 дней?	Ср 29.08.12	Чт 30.08.12	
	22	Встановлення програмного продукту	1 день?	Ср 29.08.12	Ср 29.08.12	
	23	Навчання персоналу користуванню програмний г	3 дней?	Ср 29.08.12	Пт 31.08.12	
	24	Підтримка програмного продукту	1 день?	Ср 29.08.12	Ср 29.08.12	
	25	Реалізація техпідтримки	1 день?	Ср 29.08.12	Ср 29.08.12	
	26	Випуск оновлень	1 день?	Ср 29.08.12	Ср 29.08.12	
	27	Випуск нових версій продукту	1 день?	Ср 29.08.12	Ср 29.08.12	

Наступним кроком є зв'язування завдань та задач одне за одним.

Для цього виконаємо наступні дії. У полі «Попередники» обираємо задачу, або завдання, яке було попереднім поточному.

Для цього на поточній задачі, або підзадачі на полі «Попередники» клікаємо два рази лівою кнопкою миші. Вискакує наступне вікно.



Це вікно дозволяє редагувати усі параметри, які відносяться до поточної задачі, або підзадачі.

У цьому вікні знаходимо у закладку «Попередники», й обираємо ту задачу, яка є попередньою поточній.



<b>Реалізація програмного проекту</b>	7 днів?	Ср 29.08.12	Чт 06.09.12
<b>Отримання завдання на реалізацію програмного</b>	1 день?	Ср 29.08.12	Ср 29.08.12
Зустріч з замовником	1 день	Ср 29.08.12	Ср 29.08.12
Підписання договору	1 день?	Ср 29.08.12	Ср 29.08.12
Отримання завдання	1 день?	Ср 29.08.12	Ср 29.08.12
<b>Розробка плану та концепції реалізації програмного</b>	3 днів?	Ср 29.08.12	Пт 31.08.12
Розробка концепції програмного продукту	3 днів	Ср 29.08.12	Пт 31.08.12
Затвердження концепції	1 день?	Ср 29.08.12	Ср 29.08.12
Розробка специфікації	2 днів	Ср 29.08.12	Чт 30.08.12
Розробка календарного плану			
Узгодження з замовником та затвердження			
<b>Написання програми</b>			
Вибір мови програмування			
Написання програмного продукту			
<b>Тестування</b>			
Вибір методів та систем тестування			
Тестування програмного продукту			
Обробка результатів тестування			
Усунення помилок			
<b>Впровадження</b>			
Написання файлу допомоги			
Встановлення програмного продукту			
Навчання персоналу користуванню програмою			
<b>Підтримка програмного продукту</b>			
Реалізація технічної підтримки			
Випуск оновлень			
Випуск нових версій продукту			

Сведения о задаче

Общие

Предшественники

Ресурсы

Дополнительно

Заметки

Настраиваемые поля

Название: Підписання договору
Длительность: 1д?
☒ Предв. оценка

Предшественники:

Ид.	Название задачи	Тип	Запаздывание
	Зустріч з замовником		
	Отримання завдання		
	Розробка плану та концепції реалізації програмного продукту		
	Розробка концепції програмного продукту		
	Затвердження концепції		
	Розробка специфікації		
	Розробка календарного плану		
	Узгодження з замовником та затвердження специфікації й кал		
	Написання програми		
	Вибір мови програмування		

Справка

OK

Отмена

Обираємо попередню задачу, й на діаграмі Ганта у правій половині вікна, поточна задача зв'язується з попередньою.

Файл

Формат

Сервис

Проект

Совместная работа

Окно

Справка

Нет группировки

Показать

Arial

8

Ж

К

Ч

тслеживание

Отчет

3

Название задачи	Длительность	Начало	Окончание	Предшественники	Названия рес
<b>Реалізація програмного проекту</b>	7 днів?	Ср 29.08.12	Чт 06.09.12		
<b>Отримання завдання на реалізацію програмного</b>	2 днів?	Ср 29.08.12	Чт 30.08.12		
Зустріч з замовником	1 день	Ср 29.08.12	Ср 29.08.12		
Підписання договору	1 день?	Чт 30.08.12	Чт 30.08.12	3	
Отримання завдання	1 день?	Ср 29.08.12	Ср 29.08.12		
<b>Розробка плану та концепції реалізації програмного</b>	3 днів?	Ср 29.08.12	Пт 31.08.12		
Розробка концепції програмного продукту	3 днів	Ср 29.08.12	Пт 31.08.12		

1 Авг '12

27 Авг '12

В

С

Ч

П

С

В

П

В

С

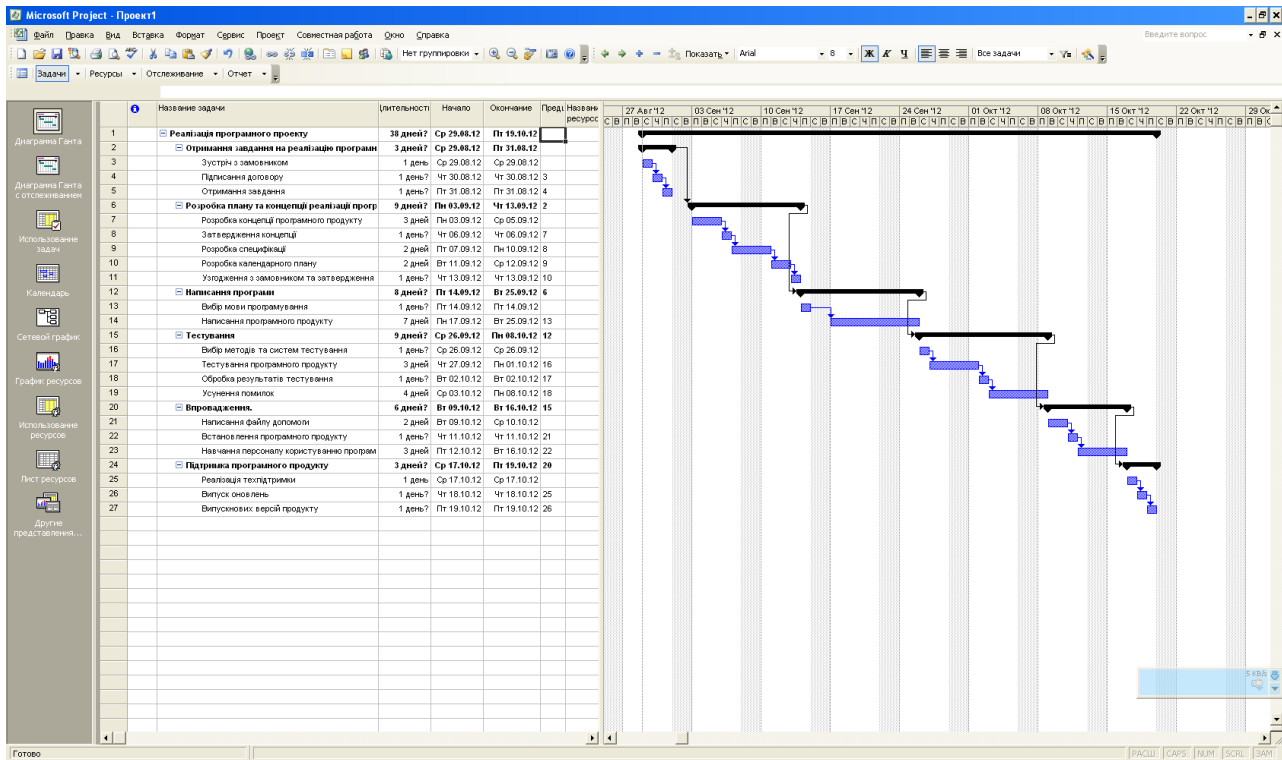
Ч

П

С

В

Виконуємо цю дію, для усіх задач, та підзадач.



На цьому Лабораторна робота № 1 вважається виконаною.

## ЗАВДАННЯ

Вибрати тему проекту, що розробляється, та побудувати, за допомогою MS-Project план виконання проекту. Повинно бути не менш ніж 6-9 завдань, у кожному з них 3-4 підзадачі.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Перелік задач та підзадач.

Реалізований календарний план у MS-Project.

Скріншот реалізованого календарного плану.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованого у MS-Project календарного плану виконання проекту.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

**КОНТРОЛЬНІ ЗАПИТАННЯ:**

1. Моделі розробки додатків?
2. Модель водоспаду?
3. Спіральна модель?
4. Універсальний процес?
5. Етапи (Універсальний процес)?
6. Фази (Універсальний процес)
7. Ітерації (Універсальний процес)
8. Компромісний трикутник
9. Принципи моделі процесу розробки
10. Версії продукту
11. Склад проектної команди
12. Модель проектної групи
13. Менеджер продукту
14. Менеджер програми
15. Розробник
16. Тестер
17. Інструктор
18. Логістик
19. Які завдання треба вирішити, щоб проект був вдалим
20. Ролі членів групи в моделі процесу розробки

## Лабораторна робота №2 (семестр 5)

**ТЕМА:** Використання ресурсів у проекті.

**МЕТА:** Отримати практичні навички в розробці плану проекту реалізації програмного продукту, чітко ставити цілі перед розробкою програми

**ЗНАТИ:** Середовище MS-Project

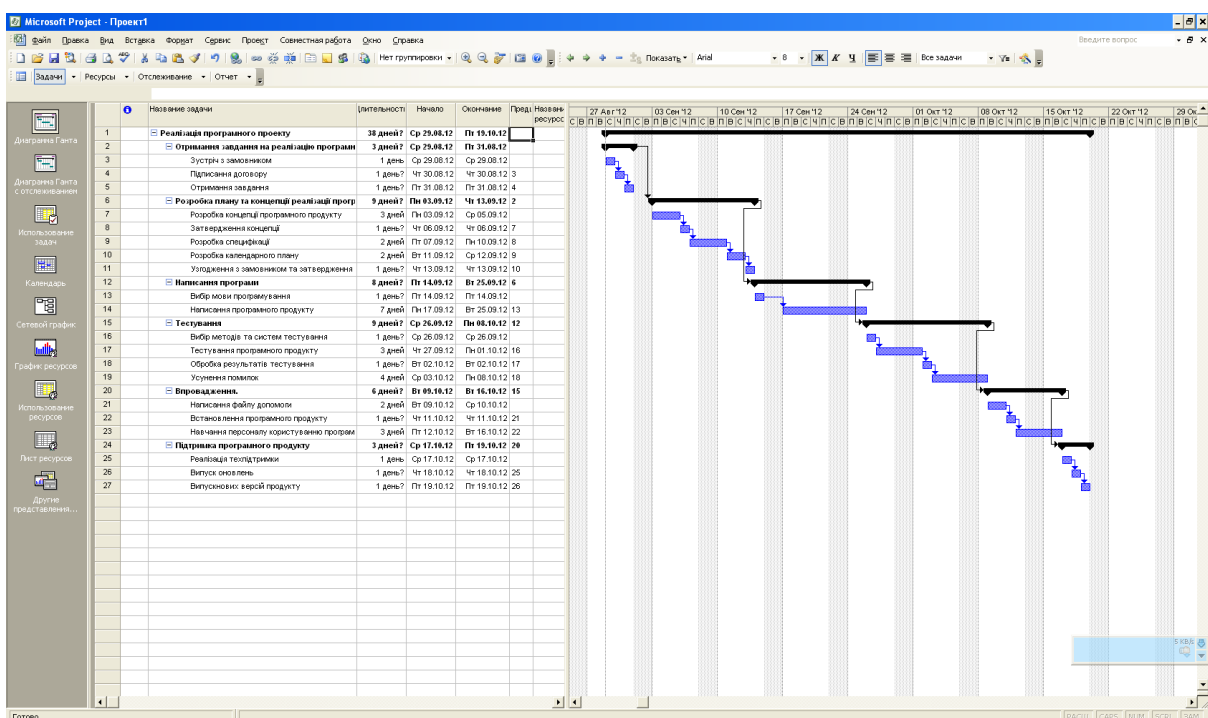
**ВМІТИ:** Інсталювати й налаштувати середовище MS-Project. Мати навички роботи з MS-Project.

**ТЕОРЕТИЧНІ ВІДОМОСТІ.** Наведені в опису лабораторної роботи №1.

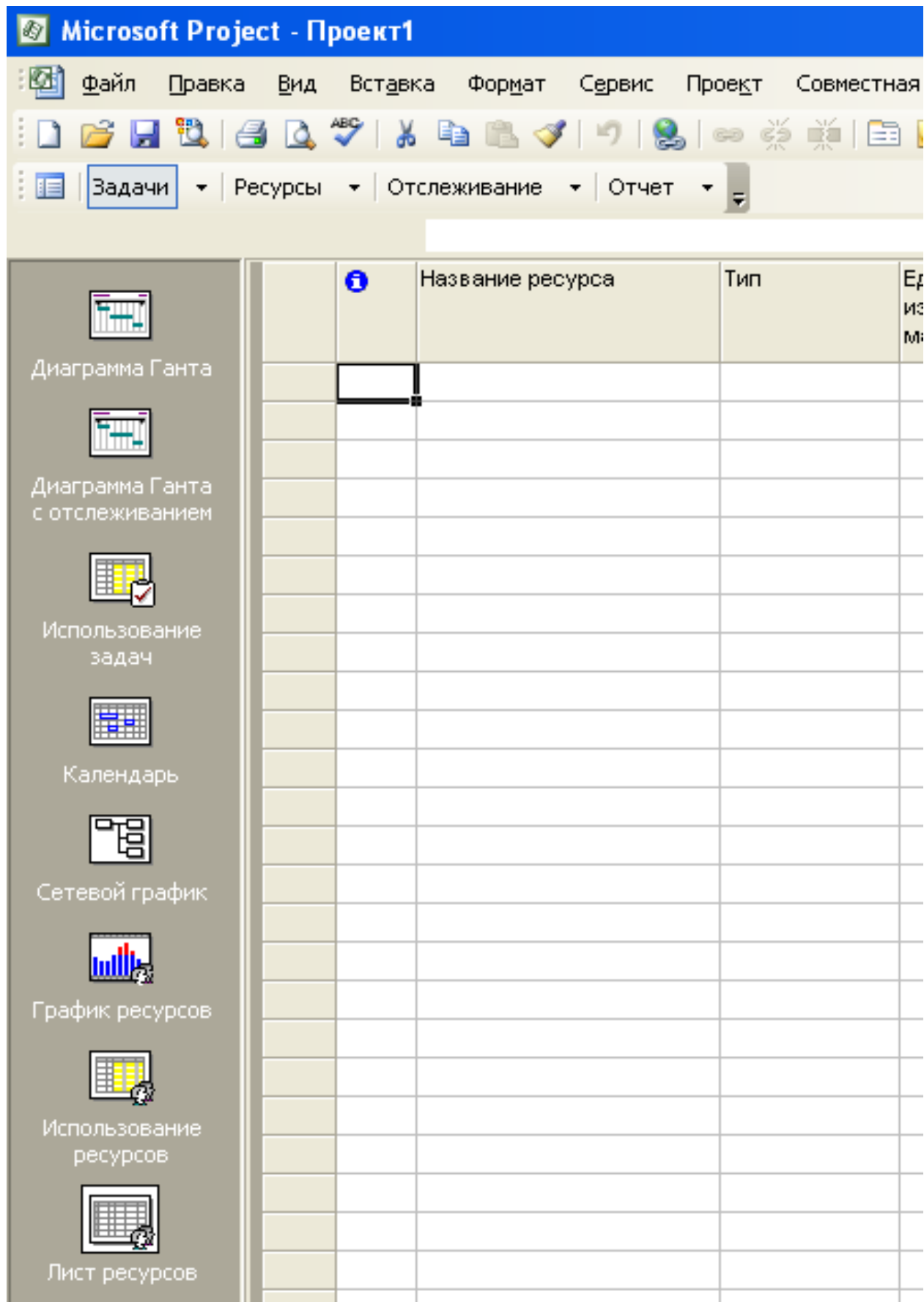
Теоретичні відомості з менеджменту проектів та створення проектних груп приведені у лекції № 1.

## ХІД ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Завантажити проект, який почали виконувати на лабораторній роботі №1.



На панелі представлень, яка знаходиться у лівій частині інтерфейсу знаходимо у панель «Лист ресурсів».



У це поле заносимо усі види ресурсів, які нам будуть потрібні, та їх вартість.

Название ресурса	Тип	Единицы измерения материалов	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Затраты на исполыз.	Начисление	Базовый календарь
Менеджер продукту	Трудовой		М		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Менеджер програми	Трудовой		М		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Розробник	Трудовой		Р		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Тестер	Трудовой		Т		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Інструктор	Трудовой		І		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Лопстик	Трудовой		Л		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Компютер	Материальный		К			0,00р.		0,00р.	Пропорциональное	
Оплата за офіс	Материальный		О			0,00р.		0,00р.	Пропорциональное	

Ресурси підрозділяються на 2 великі класи:

– трудовий – до цих ресурсів відносяться усі працівники та люди, які задіяні у процесі виконання проекту. Оплата цих ресурсів задається погодинно. (У нашому випадку це члени проектної групи)

– матеріальні – до цих ресурсів відносяться усі засоби виробництва, та інші витрати, які не пов’язані з трудовими ресурсами. (У нашому випадку це комп’ютери та витрати на утримання офісу)

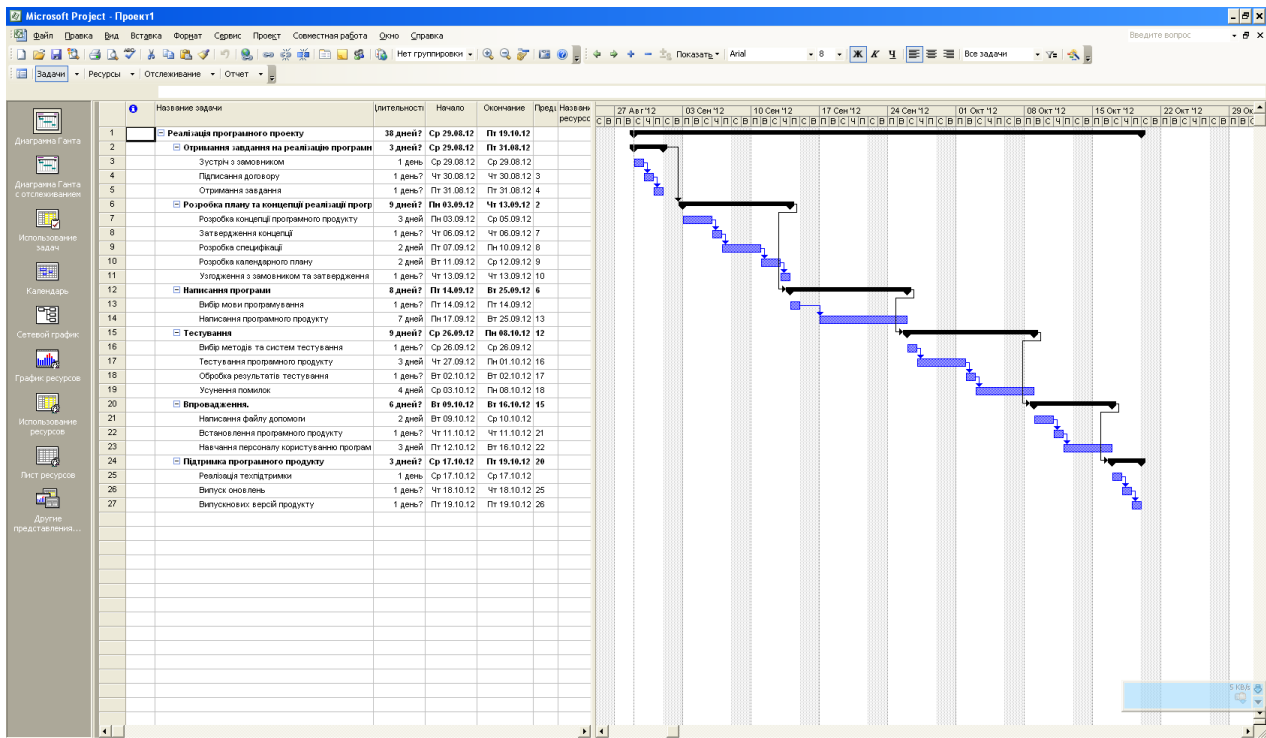
Виставимо зарплатню та витрати на одиницю ресурсу у полі «Стандартна ставка».

Название ресурса	Тип	Единицы измерения материалов	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Затраты на исполыз.	Начисление	Базовый календарь
Менеджер продукту	Трудовой		М		100%	100,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Менеджер програми	Трудовой		М		100%	75,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Розробник	Трудовой		Р		100%	50,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Тестер	Трудовой		Т		100%	50,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Інструктор	Трудовой		І		100%	50,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Лопстик	Трудовой		Л		100%	50,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Компютер	Материальный		К			5 000,00р.		0,00р.	Пропорциональное	
Оплата за офіс	Материальный		О			5 000,00р.		0,00р.	Пропорциональное	

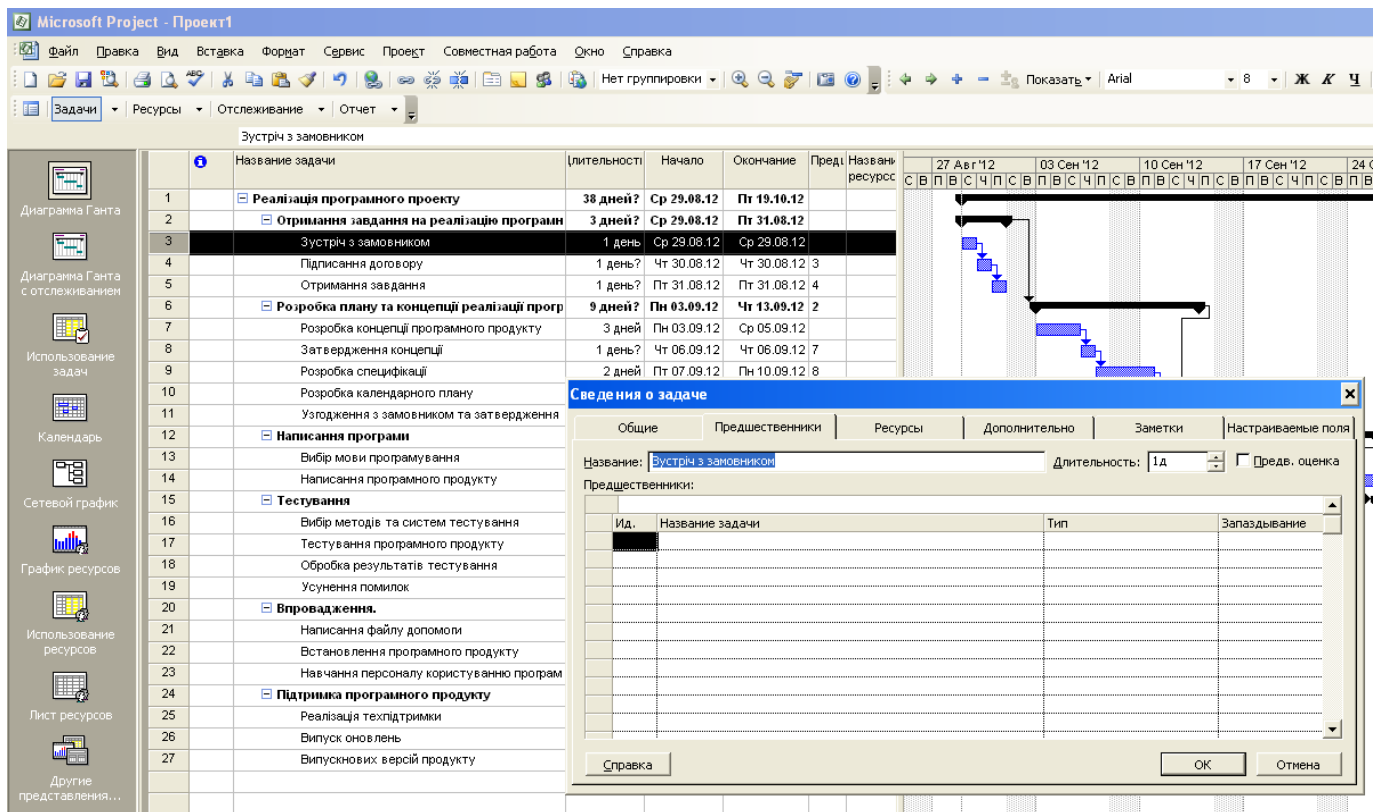
Також є можливість встановити інші параметри ресурсів відповідно до потреб.

Після того, як встановили ресурси, є необхідність прив'язати ресурси до відповідних задач.

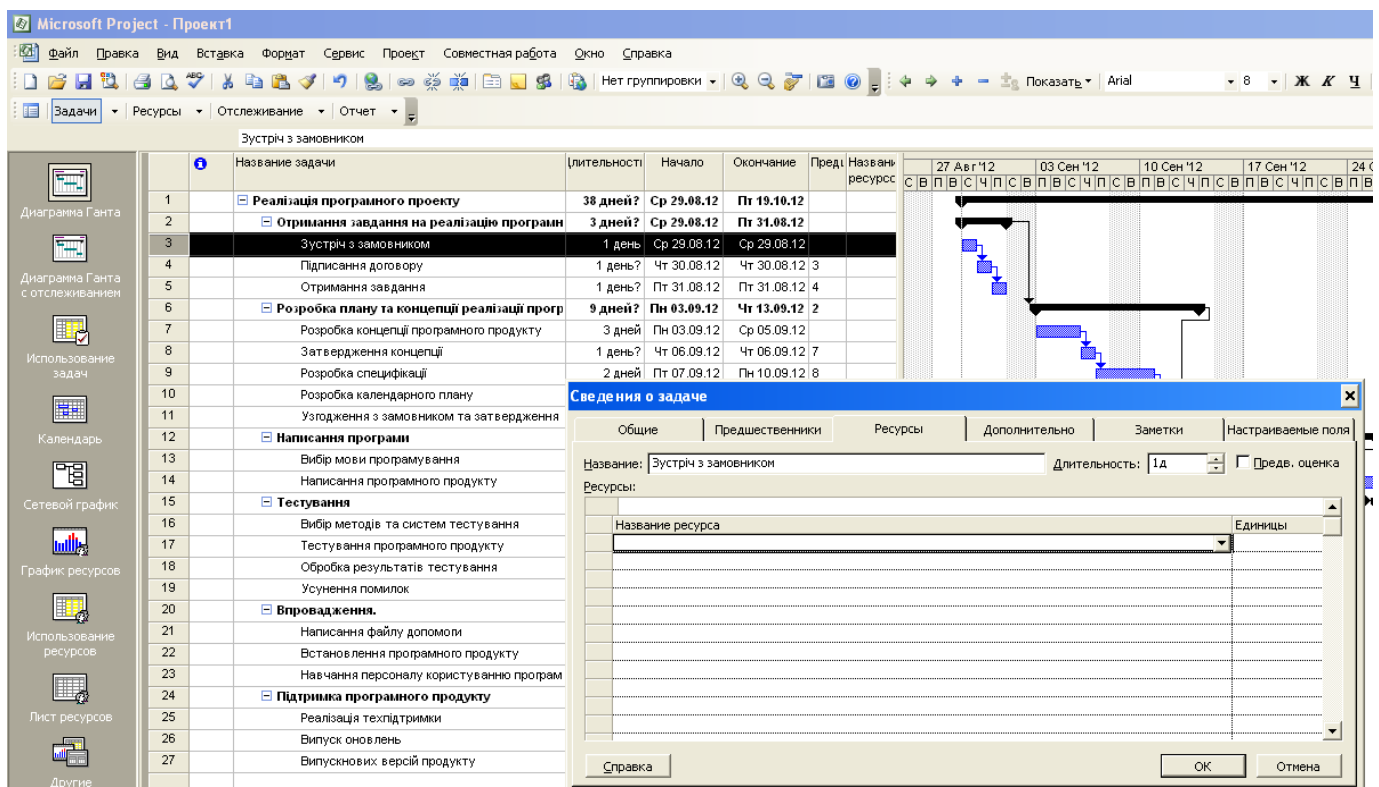
Для цього переходимо на діаграму Ганта.



Наводимо вказівник миші на кожну задачу та підзадачу, за потреби, й двічі клікаємо лівою кнопкою миші. Вискакує вікно «Відомості по задачі»



У цьому вікні заходимо у закладку «Ресурси».





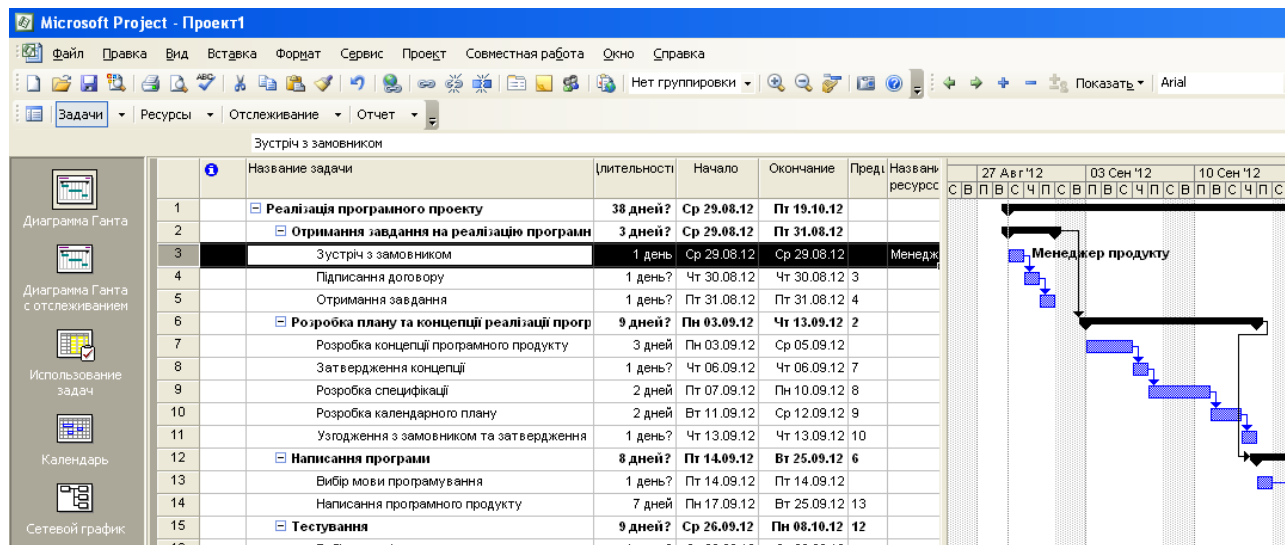
З випадаючого меню обираємо який ресурс нам потрібен, й скільки саме одиниць його нам потрібно

The screenshot shows the Microsoft Project interface with a task list on the left and a Gantt chart on the right. A dialog box titled "Сведения о задаче" (Task Information) is open, showing the "Ресурсы" (Resources) tab. The task list includes tasks such as "Реализация программного проекта", "Отримання завдання на реалізацію програм", "Зустріч з замовником", "Підписання договору", "Отримання завдання", "Розробка плану та концепції реалізації прогр", "Розробка концепції програмного продукту", "Затвердження концепції", "Розробка специфікації", "Розробка календарного плану", "Узгодження з замовником та затвердження", "Написання програми", "Вибір мови програмування", "Написання програмного продукту", "Тестування", "Вибір методів та систем тестування", "Тестування програмного продукту", "Обробка результатів тестування", "Усунення помилок", "Впровадження", "Написання файлу допомоги", "Встановлення програмного продукту", "Навчання персоналу користуванню програм", "Підтримка програмного продукту", "Реалізація технічної підтримки", "Випуск оновлень", and "Випуск нових версій продукту". The resource assignment dialog box shows a list of resources: "Менеджер продукту", "Менеджер програми", "Розробник", "Тестер", and "Інструктор".

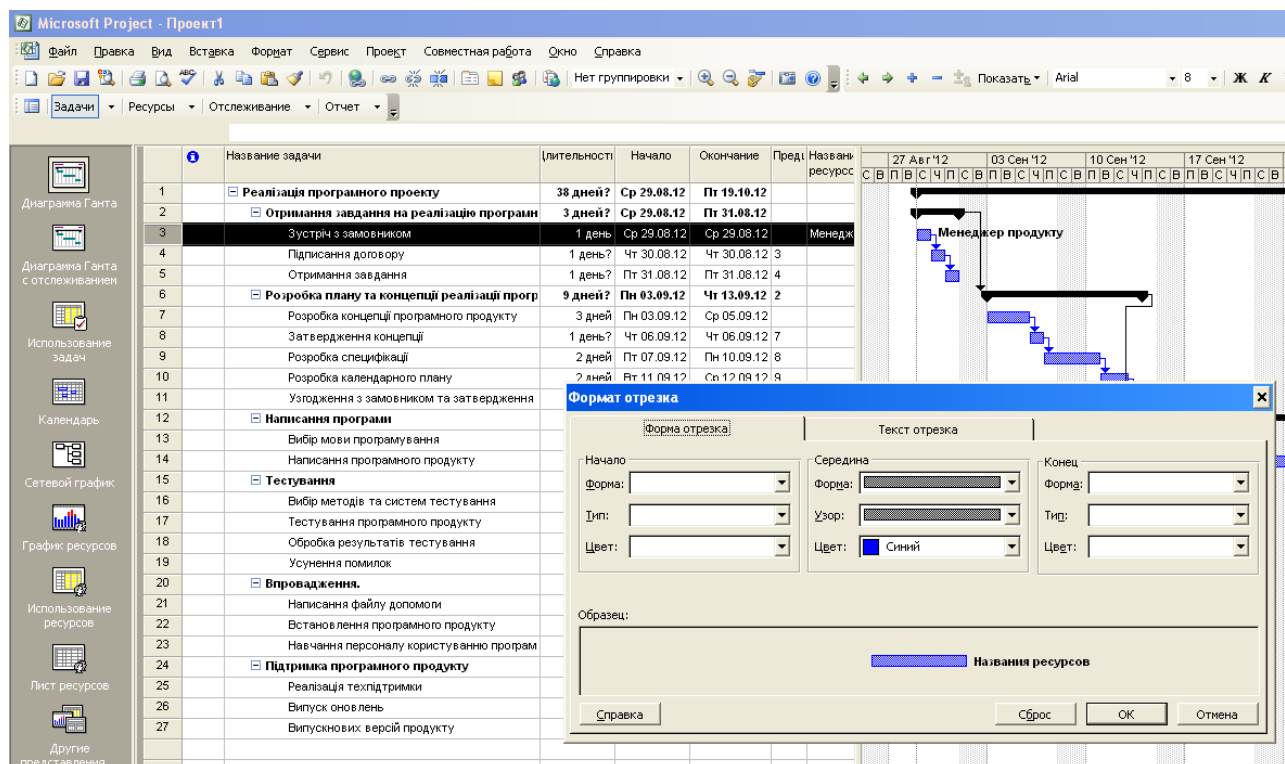
The screenshot shows the Microsoft Project interface with a task list on the left and a Gantt chart on the right. A dialog box titled "Сведения о задаче" (Task Information) is open, showing the "Ресурсы" (Resources) tab. The task list is the same as in the previous screenshot. The resource assignment dialog box shows a list of resources: "100%", "Менеджер продукту", and "Менеджер програми". The "Менеджер продукту" resource is assigned to the task "Зустріч з замовником" with a value of 100%.

Для трудовых ресурсов 1 человека равноюс 100%. Для материальных ресурсов одна единица техники (например) теж дорівнюе 100%.

На діаграмі Ганта відображається ресурси, які ми виділили на вирішення задачі.



Щоб визначити скільки грошей потрібно для виконання задачі, наводимо мишу на відрізок на діаграмі Ганта й двічі клікаємо мишкою.



Переходимо у закладку «Текст відрізка».

Microsoft Project - Проект1

Файл Правка Вид Вставка Формат Сервис Проект Совместная работа Окно Справка

Задачи Ресурсы Отслеживание Отчет

Название задачи [длительности] Начало Окончание Пред. Назван ресурс

1 Реализация программного проекта 38 дней? Ср 29.08.12 Пт 19.10.12

2 Отримання завдання на реалізацію програм 3 днів? Ср 29.08.12 Пт 31.08.12

3 Зустріч з замовником 1 день? Ср 29.08.12 Ср 29.08.12 Менедж

4 Підписання договору 1 день? Чт 30.08.12 Чт 30.08.12 3

5 Отримання завдання 1 день? Пт 31.08.12 Пт 31.08.12 4

6 Розробка плану та концепції реалізації прог 9 днів? Пн 03.09.12 Чт 13.09.12 2

7 Розробка концепції програмного продукту 3 днів? Пн 03.09.12 Ср 05.09.12

8 Затвердження концепції 1 день? Чт 06.09.12 Чт 06.09.12 7

9 Розробка специфікації 2 днів? Пт 07.09.12 Пн 10.09.12 8

10 Розробка календарного плану 2 днів? Пт 11.09.12 Ср 12.09.12 9

11 Узгодження з замовником та затвердження

12 Написання програми

13 Вибір мови програмування

14 Написання програмного продукту

15 Тестування

16 Вибір методів та систем тестування

17 Тестування програмного продукту

18 Обробка результатів тестування

19 Усунення помилок

20 Впровадження.

21 Написання файлу допомоги

22 Встановлення програмного продукту

23 Навчання персоналу користуванню програм

24 Підтримка програмного продукту

25 Реалізація техпідтримки

26 Випуск оновлень

27 Випуск нових версій продукту

Формат отрезка

Форма отрезка Текст отрезка

Слева Названия ресурсов

Справа

Сверху

Снизу

Внутри

Образец:

Названия ресурсов

Справка Сброс ОК Отмена

Й обираємо з якої сторони будемо показувати затрати на використання ресурсу у даній задачі (у нашому прикладі оберемо знизу).

Microsoft Project - Проект1

Файл Правка Вид Вставка Формат Сервис Проект Совместная работа Окно Справка

Задачи Ресурсы Отслеживание Отчет

Название задачи [длительности] Начало Окончание Пред. Назван ресурс

1 Реализация программного проекта 38 дней? Ср 29.08.12 Пт 19.10.12

2 Отримання завдання на реалізацію програм 3 днів? Ср 29.08.12 Пт 31.08.12

3 Зустріч з замовником 1 день? Ср 29.08.12 Ср 29.08.12 Менедж

4 Підписання договору 1 день? Чт 30.08.12 Чт 30.08.12 3

5 Отримання завдання 1 день? Пт 31.08.12 Пт 31.08.12 4

6 Розробка плану та концепції реалізації прог 9 днів? Пн 03.09.12 Чт 13.09.12 2

7 Розробка концепції програмного продукту 3 днів? Пн 03.09.12 Ср 05.09.12

8 Затвердження концепції 1 день? Чт 06.09.12 Чт 06.09.12 7

9 Розробка специфікації 2 днів? Пт 07.09.12 Пн 10.09.12 8

10 Розробка календарного плану 2 днів? Пт 11.09.12 Ср 12.09.12 9

11 Узгодження з замовником та затвердження

12 Написання програми

13 Вибір мови програмування

14 Написання програмного продукту

15 Тестування

16 Вибір методів та систем тестування

17 Тестування програмного продукту

18 Обробка результатів тестування

19 Усунення помилок

20 Впровадження.

21 Написання файлу допомоги

22 Встановлення програмного продукту

23 Навчання персоналу користуванню програм

24 Підтримка програмного продукту

25 Реалізація техпідтримки

26 Випуск оновлень

27 Випуск нових версій продукту

Формат отрезка

Форма отрезка Текст отрезка

зат

Слева Названия ресурсов

Справа

Сверху

Снизу

Внутри

Образец:

Затраты на сверхурочные

Затраты1

Затраты2

Затраты3

Затраты4

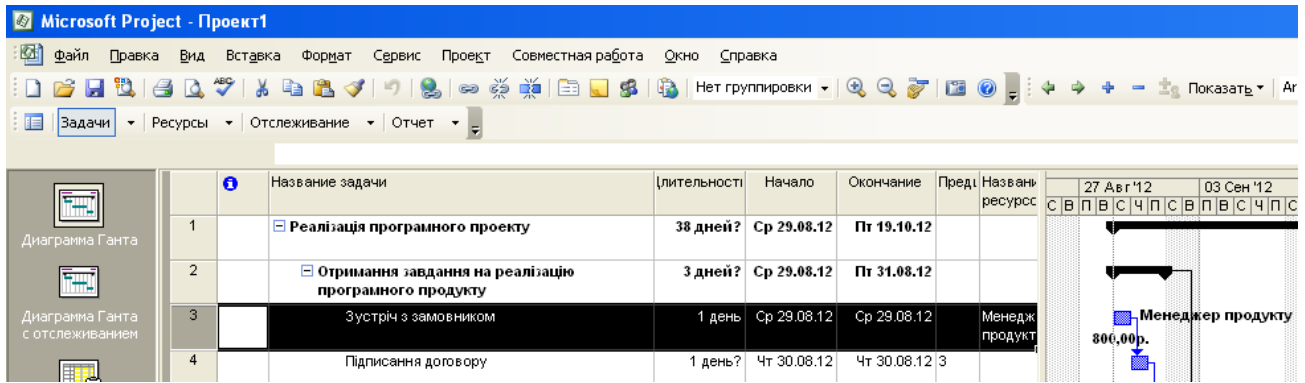
Затраты5

Затраты6

Затраты7

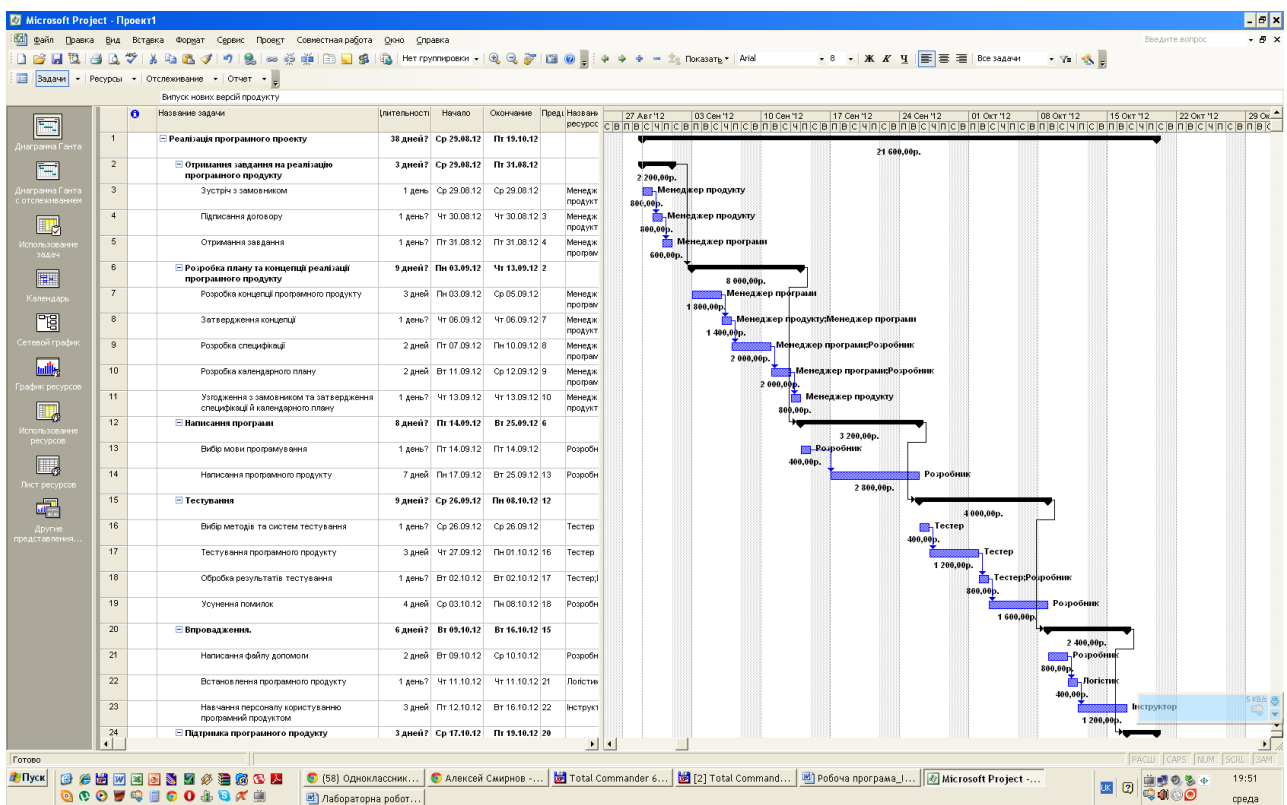
Справка Сброс ОК Отмена

MS-Project сам автоматично підрахує затрати.



Далі у лабораторній роботі для усіх завдань виставляємо ресурси та розрахунок затрат.

Таким чином реалізований проект повинен мати наступний вигляд:



На цьому Лабораторна робота № 2 вважається виконаною.

## **ЗАВДАННЯ.**

Реалізувати календарний план з використання ресурсів у MS-Project.

Оцінити правдоподібність результатів

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Перелік задач та підзадач.

Реалізований календарний план з використання ресурсів у MS-Project.

Оцінка правдоподібності результатів

Скріншот реалізованого календарного плану з використання ресурсів .

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованого у MS-Project календарного плану виконання проекту.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче).

## **КОНТРОЛЬНІ ЗАПИТАННЯ:**

1. Моделі розробки додатків?
2. Модель водоспаду?
3. Спіральна модель?
4. Універсальний процес?
5. Етапи (Універсальний процес)?
6. Фази (Універсальний процес)
7. Ітерації (Універсальний процес)
8. Компромісний трикутник

9. Принципи моделі процесу розробки
10. Версії продукту
11. Склад проектної команди
12. Модель проектної групи
13. Менеджер продукту
14. Менеджер програми
15. Розробник
16. Тестер
17. Інструктор
18. Логістик
19. Які завдання треба вирішити, щоб проект був вдалим
20. Ролі членів групи в моделі процесу розробки

## **Лабораторна робота №3 (семестр 5)**

**ТЕМА:** Написання специфікації на програму, що розробляється

**МЕТА:** Отримати практичні навички в розробці специфікацій програм, чітко ставити цілі перед розробкою програми

**ЗНАТИ:** Мова програмування C або Pascal

**ВМІТИ:** Інсталювати RAD (наприклад Builder або Delphi). Знати основи розробки програм під ОС Windows. Уміти користуватися текстовим редактором.

**ТЕОРЕТИЧНІ ВІДОМОСТІ.** Теоретичні відомості наведені у лекції № 2.

У зв'язку з великим об'ємом інформації використовувати електронну документацію (погоджувати з лектором).

### **ЗАВДАННЯ.**

1. Вибрати тему програми, що розробляється, з напрямів: системні утиліти; мультимедіа; ігрова програма; графіка; Інтернет; офіс.

Примітка. Можливий вибір теми вже розробленої програми для її подальшої модифікації в наступних випадках:

– Коли програма була реалізована студентом як курсовий проект, курсова робота або самостійна робота на кафедрі;

– Коли студент брав участь в розробці програми або зацікавлений в модифікації чужої програми. Якщо ліцензія поширення програми (Freeware) дозволяють її використання.

2. У текстовому виді розробити специфікацію програми (див. приклад порядку виконання роботи).

## **ПРИКЛАД ПОРЯДКУ ВИКОНАННЯ РОБОТИ.**

Специфікація програми повинна відповідати наступному змісту.

### **1. Загальний опис**

#### **1.1 Вхідні данні**

#### **1.2 Вихідні дані**

#### **1.3 Опис файлів, що входять в пакет**

### **2. Опис архітектури**

### **3. Функціональні вимоги**

**ЯК ПРИКЛАД РОЗГЛЯНЕМО СПЕЦИФІКАЦІЮ ПРОГРАМИ "КАЛЬКУЛЯТОР".**

### **1. Загальний опис**

Калькулятор складається з трьох модулів – "Графічний інтерфейс", "Модуль, що аналізує і обчислює введенний вираз" (AnalaizerClass.dll) і "Модуль, що реалізовує математичні функції" (CalcClass.dll). Після того, як користувач введе обчислюване вираження одним з двох вищеописаних способів, управління передається аналізуючому модулю, який форматує вираз, виділяючи числа і оператори, перевіряє коректність дужкової структури, а також виявляє невірні з точки зору математики конструкції (наприклад,  $3+*+3$ ), переводить вираз в зворотний польський запис, після чого обчислює вирази, використовуючи математичні функції з модуля CalcClass. Користувач також може користуватися довідковою системою для уточнення тонкощів роботи програми.

### **2. Опис інтерфейсу**

#### **2.1. Вхідні дані**

##### **2.1.1. Параметри виклику (формат командного рядка)**

calc.exe [expression]

expression – математичний вираз, що задовольняє вимозі 3.2

##### **2.1.2. Стан інформаційного оточення**

У папці з програмою також знаходяться файли CalcClass.dll, AnalaizerClass.dll



## **2.2. Вихідні дані**

### **2.2.1. Коди повернення програми**

Число і 0 на новому рядку – результат обчислення виразу.

Error: <повідомлення про помилку> і код помилки на новому рядку – повідомлення про помилку у разі невідповідності вхідного виразу вимогам 3.2

### **2.2.2. Стан інформаційного оточення після завершення програми**

У папці з програмою також знаходяться файли CalcClass.dll, AnalizerClass.dll

### **2.2.3. Повідомлення про помилки, що видаються програмою (коди помилок)**

Error 01 at <i> – неправильна дужкова структура, помилка на <i> символі

Error 02 at <i> – невідомий оператор на <i> символі.

Error 03 – невірна синтаксична конструкція вхідного виразу

Error 04 at <i> – Два підряд оператора на <i> символі.

Error 05 – Незавершений вираз.

Error 06 – Занадто мале або занадто велике значення числа для int.

Числа мають бути в межах від – 2147483648 до 2147483647

Error 07 – Занадто довгий вираз. Максимальна довжина – 65536 символів.

Error 08 – Сумарна кількість чисел і операторів перевищує 30

Error 09 – Помилка ділення на 0.

## **2.3. Опис файлів, що входять в пакеті калькулятора**

CalcClass.dll – бібліотека, в якій реалізовані усі необхідні математичні функції.

AnalizerClass.dll – модуль, в якому реалізований синтаксичний розбір виразу, а також його обчислення.

calc.exe – графічна оболонка, головний модуль.

Calc.hlp – довідкова система програми

### 3. Опис архітектури

Як вже відзначалося вище, в архітектурі системи виділено 3 модулі. Кожен з модулів займається певним завданням. Відповідно, Система – це взаємодія цих 3-х модулів. Розглянемо їх детальніше.

#### 1. Модуль математичних операцій (CalcClass.dll)

Модуль містить усі математичні функції, використовувані в програмі.

```
/// <summary>
    /// Функція складання числа a і b
/// </summary>
/// <param name="a">доданок</param>
/// <param name="b">доданок</param>
/// <returns>сума</returns>
public static int Add(long a, long b)
/// <summary>
    /// функція віднімання чисел a і b
/// </summary>
/// <param name="a">зменшуване</param>
/// <param name="b">від'ємник</param>
/// <returns>різниця</returns>
public static int Sub(long a, long b)
/// <summary>
    /// функція множення чисел a і b
/// </summary>
/// <param name="a">множник</param>
/// <param name="b">множник</param>
/// <returns>добуток</returns>
public static int Mult(long a, long b)
/// <summary>
    /// функція знаходження частки
/// </summary>
```

```
/// <param name="a">ділиме</param>
/// <param name="b">діленьник</param>
/// <returns>частка</returns>
```

```
public static int Div(long a, long b)
```

```
/// <summary>
```

```
/// функція ділення по модулю
```

```
/// </summary>
```

```
/// <param name="a">ділиме</param>
```

```
/// <param name="b">діленьник</param>
```

```
/// <returns>залишок</returns>
```

```
public static int Mod(long a, long b)
```

```
/// <summary>
```

```
/// унарний плюс
```

```
/// </summary>
```

```
/// <param name="a"></param>
```

```
/// <returns></returns>
```

```
public static int ABS(long a)
```

```
/// <summary>
```

```
/// унарний мінус
```

```
/// </summary>
```

```
/// <param name="a"></param>
```

```
/// <returns></returns>
```

```
public static int IABS(long a)
```

Використовується також глобальна змінна:

```
/// <summary>
```

```
/// Останнє повідомлення про помилку.
```

```
/// Поле і властивість для нього
```

```
/// </summary>
```

```
private static string _lastError = "";
```

```
public static string lastError
```

*Модуль аналізу і обчислення виразів*

Складається з наступних методів і властивостей :

```
/// <summary>
/// позиція виразу, на якій найдена
/// синтаксична помилка (у разі знаходження на етапі
/// виконання – не визначається)
/// </summary>
private static int erposition = 0;

/// <summary>
/// Вхідний вираз
/// </summary>
public static string expression = "";

/// <summary>
/// Показує, чи є необхідність у виведенні
/// повідомлень про помилки. У разі консольного запуску
/// програми це значення – false.
/// </summary>
public static bool ShowMessage = true;

/// <summary>
/// Перевірка коректності дужкової структури вхідного
/// виразу
/// </summary>
/// <returns>true – якщо усе нормально, false – якщо є
/// помилка</returns>
/// метод біжить по вхідному виразу, символ за
/// символом аналізуючи його і рахуючи кількість
/// дужок. У разі виникнення помилки повертає false
/// а в erposition записує позицію, на якій виникла
/// помилка.
public static bool CheckCurrency()
```

```

    /// <summary>
    /// Форматує вхідний вираз, виставляючи між
    /// операторами пропуски і видаляючи зайві, а також
    /// відловлює нерозпізнані оператори, стежить за кінцем
    /// рядка а також відловлює помилки на кінці рядка
    /// </summary>

    /// <returns>кінцевий рядок або повідомлення про
    /// помилку, що починаються із спец. символу &

</returns>

public static string Format()
    /// <summary>
    /// Створює масив, в якому розташовуються оператори і
    /// символи, представлені в зворотному польському записі
    /// (бездужковий)
    /// На цьому ж етапі відловлюються майже усі інші
    /// помилки (див. код). По суті – це компіляція.
    /// </summary>
    /// <returns>масив зворотного польського
    /// запису</returns>

public static System.Collections.ArrayList CreateStack()
    /// <summary>
    /// Обчислення зворотного польського запису
    /// </summary>
    /// <returns>результат обчислень або повідомлення про
    /// помилку</returns>

public static string RunEstimate()
    /// <summary>
    /// Метод, організуючий обчислення. По черзі запускає
    /// CheckCorrncy, Format, CreateStack і RunEstimate
    /// </summary>

```

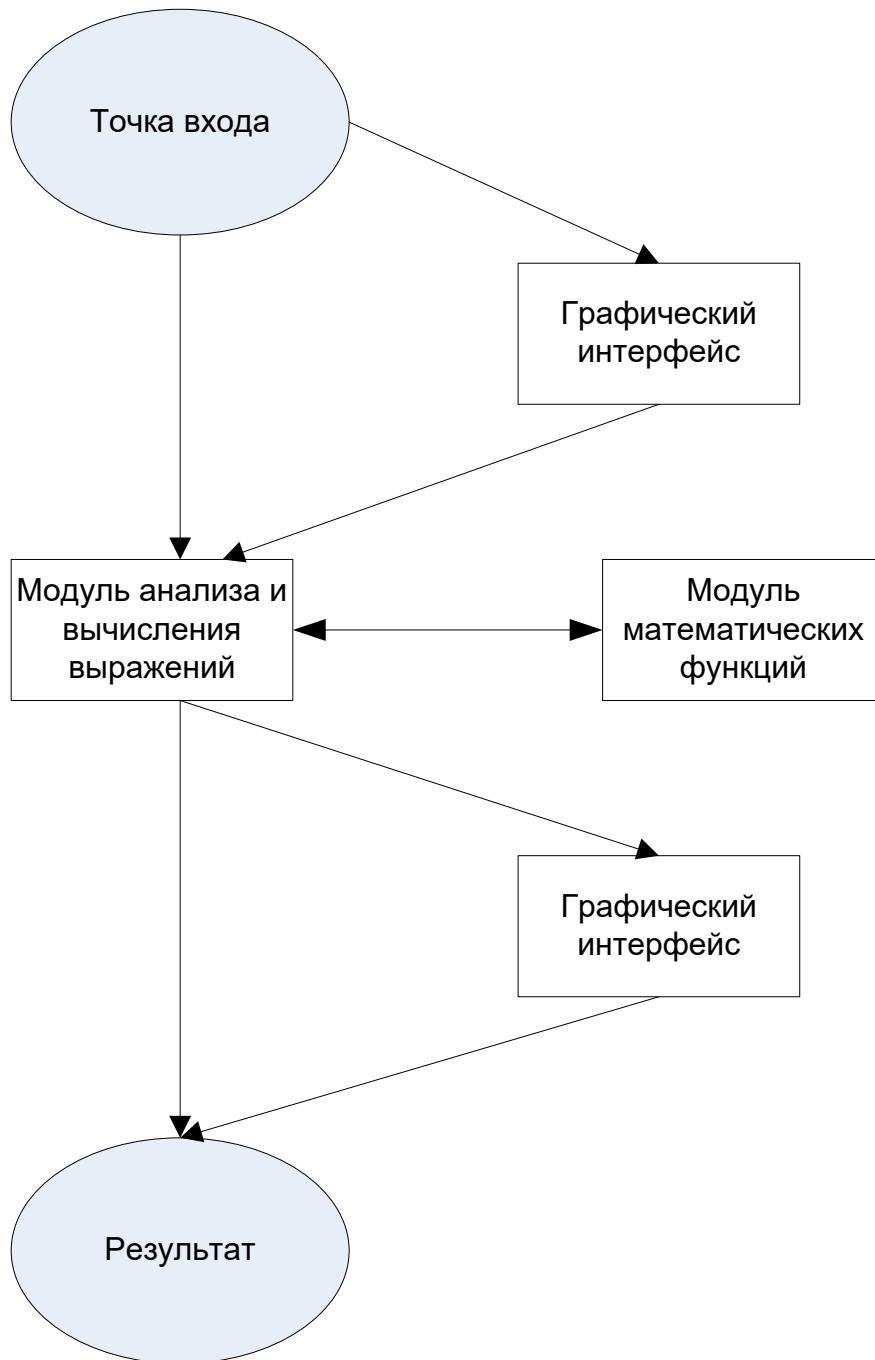
/// <returns></returns>

public static string Estimate()

*Модуль графічного інтерфейсу – забезпечує управління системи в графічній формі. Основні функції цього модуля – введення і виведення даних.*

*Функціональні вимоги*

Взаємодія модулів показана на малюнку:



### 3. Функціональні вимоги

#### 3.1. Вимоги до програми

- 3.1.1. Калькулятор повинен виконувати наступні арифметичні операції
- 3.1.2. складання, віднімання, множення, знаходження частки, знаходження залишку. Специфікацію на них см 3.2.
- 3.1.3. Калькулятор повинен підтримувати роботу з цілими числами в
- 3.1.4. межах від  $-2147483648$  до  $2147483647$  (надалі MININT і MAXINT). У разі виходу за ці межі повинне видаватися повідомлення про помилку Error 06.
- 3.1.5. Калькулятор повинен мати пам'ять на одне ціле число, а також
- 3.1.6. можливість виводити це число на екран, скидати його значення на 0 і додавати до нього будь-яке інше число, введене в поле введення.
- 3.1.7. При натисненні на клавішу M+ до числа, записаного в
- 3.1.8. пам'ять, додається число, записане в полі "Результат". При цьому на складання накладаються обмеження з 3.2.1.
- 3.1.9. Якщо в полі "Результат" записаний код помилки, то при натисненні на клавішу M+ повинне видаватися повідомлення "Неможливо перетворити до числа".
- 3.1.10. При натисненні на кнопку MC число в пам'яті обнуляється.
- 3.1.11. При натисненні на кнопку MR число з пам'яті приписується в кінець виразу в рядку "Вираз".
- 3.1.12. Калькулятор повинен надавати можливість користувачеві працювати з операціями унарного плюса і унарного мінуса.
- 3.1.13. Якщо між натисненнями на кнопку  $<+/->$  проходить менше 3 секунд, то введений оператор міняється на протилежного.
- 3.1.14. Якщо між натисненнями на кнопку  $<+/->$  проходить більше 3 секунд, то до вираження дописується знак "-".
- 3.1.15. Калькулятор повинен мати графічний інтерфейс, що містить кнопки з цифрами і арифметичними операціями, кнопкою рівності, кнопками роботи з пам'яттю, кнопками редагування дужок і

кнопками скидання, перемикачем унарного мінуса/унарного плюса, текстовими полями для введення виразу і виведення результату.

**3.1.16.** При натисненні на клавішу <Enter> калькулятор повинен проводити обчислення виразу.

**3.1.17.** При натисненні на клавішу <ESC> програма повинна припиняти свою роботу.

**3.1.18.** У разі невірно побудованого обчислюваного виразу або невідповідності його вимогам 3.2 в текстове вікно результат повинно виводитися відповідні повідомлення (см 2.2.3)

**3.1.19.** Арифметичні операції

**3.1.20.** Складання

**3.1.21.** Для чисел, кожне з яких менше або рівне MAXINT і більше або рівне MININT, функція підсумовування повинна повертати правильну суму з точки зору математики.

**3.1.22.** Для чисел, сума яких більше ніж MAXINT і менше ніж MININT, а також у разі, якщо будь-який з доданків більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06(см 2.2.3).

**3.1.23.** Віднімання

**3.1.24.** Для чисел, кожне з яких менше або рівне MAXINT і більше або рівне MININT, функція віднімання повинна повертати правильну різницю з точки зору математики.

**3.1.25.** Для чисел, різниця яких більше ніж MAXINT і менше ніж MININT, а також у разі, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06(см 2.2.3).

**3.1.26.** Множення

**3.1.27.** Для чисел, добуток яких менше або рівне MAXINT і більше або рівне MININT, функція множення повинна повертати правильний добуток з точки зору математики.



- 3.1.28.** Для чисел, добуток яких більше ніж MAXINT і менше ніж MININT, а також у разі, якщо будь-який з множників більше ніж MAXINT або менше ніж MININT, програма повинна видавати помилку Error 06(см 2.2.3).
- 3.1.29.** Знаходження частки
- 3.1.30.** Для чисел, менших або рівних MAXINT і великих або рівних MININT, частка яких менше або рівне MAXINT і більше або рівне MININT і дільник не рівний 0, функція ділення повинна повертати правильну частку з точки зору математики.
- 3.1.31.** Для чисел, частка яких більше ніж MAXINT і менше ніж MININT, а також у разі, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, і для дільника, не рівного 0, програма повинна видавати помилку Error 06(см 2.2.3).
- 3.1.32.** Якщо дільник дорівнює 0, програма повинна видавати помилку Error 09.
- 3.1.33.** Ділення із залишком
- 3.1.34.** Для чисел, менших або рівних MAXINT і більших або рівних MININT, залишок яких менше або рівний MAXINT і більше або рівний MININT і дільник не рівний 0, функція ділення повинна повертати правильний залишок з точки зору математики.
- 3.1.35.** Для чисел, залишок яких більше ніж MAXINT і менше ніж MININT, а також у разі, якщо будь-яке з чисел більше ніж MAXINT або менше ніж MININT, і для дільника, не рівного 0, програма повинна видавати помилку Error 06(см 2.2.3).
- 3.1.36.** Якщо дільник дорівнює 0, програма повинна видавати помилку Error 09.
- 3.1.37.** Унарний плюс \ мінус
- 3.1.37.1.** Для чисел, менших або рівних MAXINT і великих або рівних MININT, операція унарного плюса / мінуса повинна повертати число відповідного знаку.

- 3.1.38.** Для чисел, великих MAXINT або менших MININT, функція повинна видавати помилку Error 06(см 2.2.3).
- 3.1.39.** Додаткові вимоги до вхідного вираження
- 3.1.40.** Максимальне сумарне число операторів і чисел – 30.
- 3.1.41.** Максимальна глибина вкладеності дужкової структури – 3.
- 3.1.42.** Як унарний мінус використовується символ "m", як унарний плюс – "p".
- 3.1.43.** Для операції знаходження частки – "/", для знаходження залишку – "mod".
- 3.1.44.** Між операторами дужками і числами може бути будь-яка кількість пропусків.
- 3.1.45.** Дозволяється використовувати лише дужки виду "(" і ")".
- 3.1.46.** Максимальна довжина виразу – 65535 символів.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Реалізована специфікація

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованої специфікації.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

## **КОНТРОЛЬНІ ЗАПИТАННЯ:**

1. Програмне забезпечення.
2. Види програмного забезпечення
3. Розробка програмного забезпечення
4. Класи програмного забезпечення
5. Розробка вимог до програмної системи
6. Види вимог до програмного забезпечення за рівнями
7. Види вимог до програмного забезпечення за характером
8. Джерела вимог до програмного забезпечення
9. Методи знаходження вимог до програмного забезпечення
10. Документування вимог до програмного забезпечення
11. Програмна документація
12. Види програмних документів
13. Специфікація вимог до програмного забезпечення
14. Прецеденти
15. Загальний план специфікації вимог до ПЗ
15. Вимоги до зовнішніх інтерфейсів у специфікації
16. Атрибути програмного продукту

## Лабораторна робота №4 (семестр 5)

**ТЕМА:** Моделювання інформаційних систем з використанням CASE засобів.

**МЕТА:** Знайомство з уніфікованою мовою моделювання (UML), знайомство з пакетом візуального моделювання Visio 2007 і отримання навичок моделювання конкретної інформаційної системи в CASE системі.

**ЗНАТИ:** Мову програмування С або Pascal.

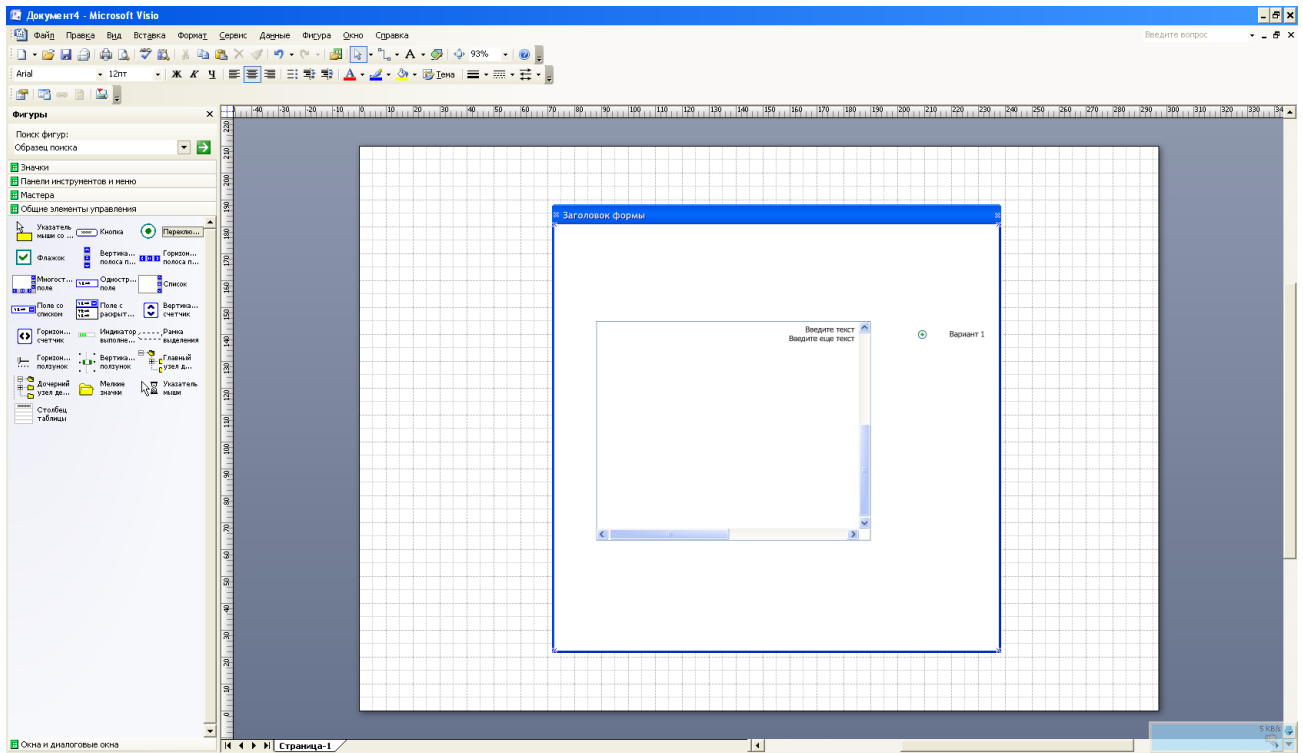
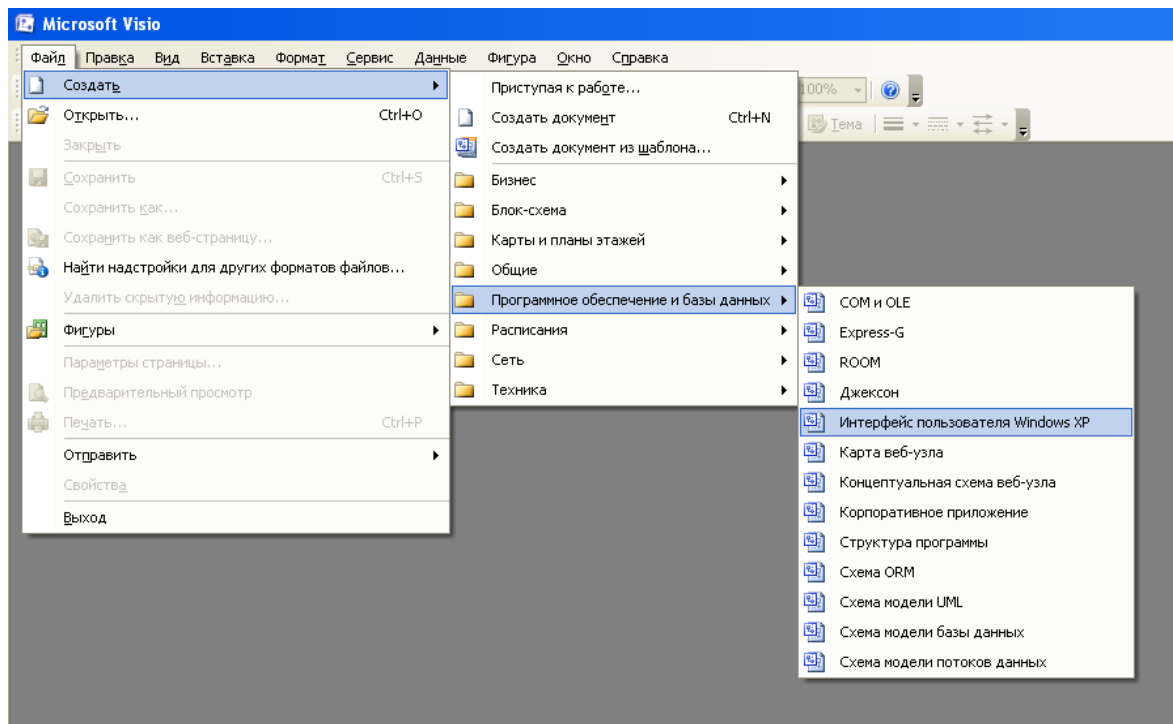
**ВМІТИ:** Працювати з Microsoft Visio 2007.

**ТЕОРЕТИЧНІ ВІДОМОСТІ.** Теоретичні відомості наведені у лекції № 3.

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

### ЗАВДАННЯ:

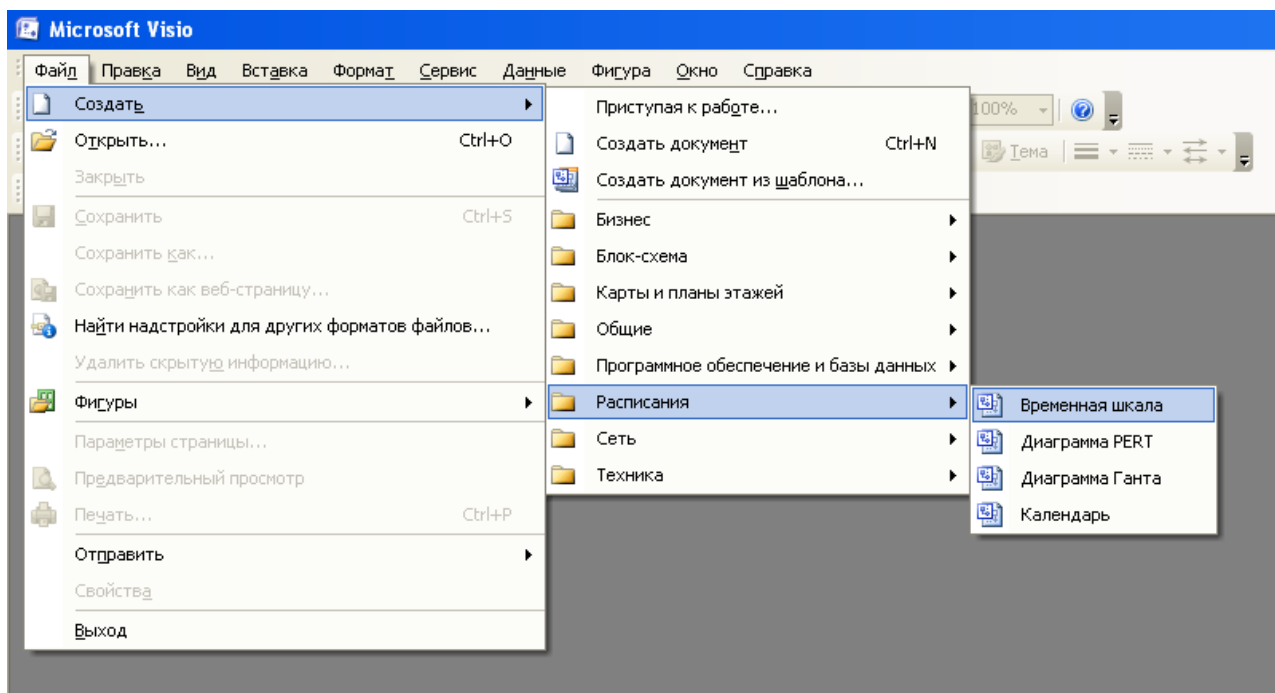
1. На основі розробленої специфікації в ЛР № 3. У пакеті візуального моделювання Visio 2007 розробити інтерфейс користувача. (Файл> Створити> Програмне забезпечення та бази даних> Інтерфейс користувача Windows XP). Інтерфейс створюється для візуалізації зовнішнього вигляду програми.



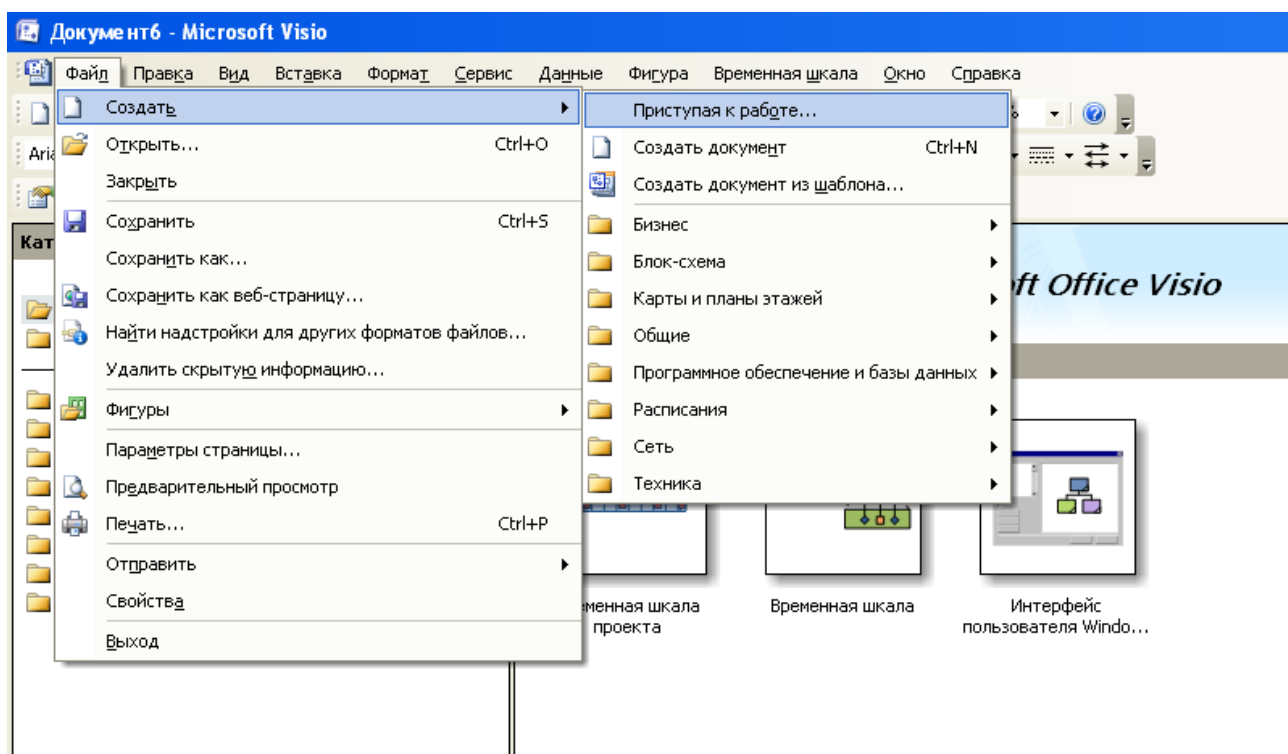
## 2. Створити графік розробки і запуску продукту.

Проект необхідно реалізовувати той, який був зроблений у ході виконання лабораторних робіт № 1-2.

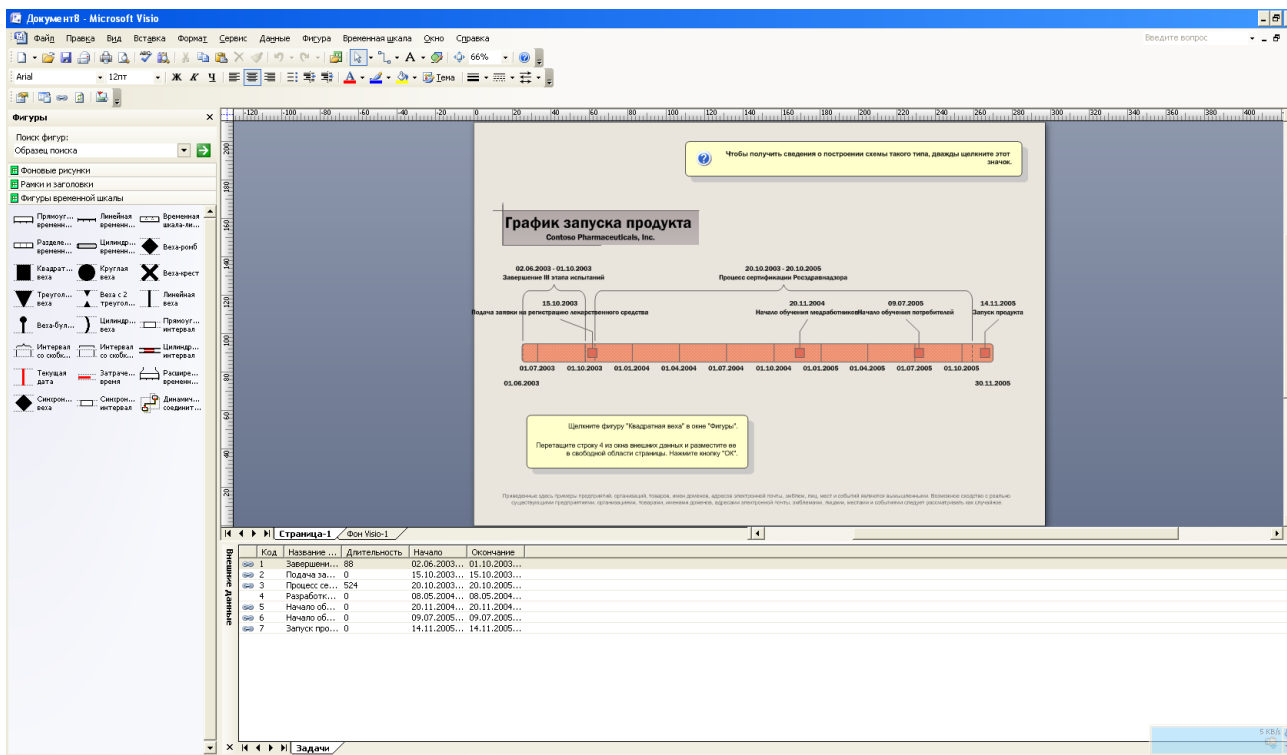
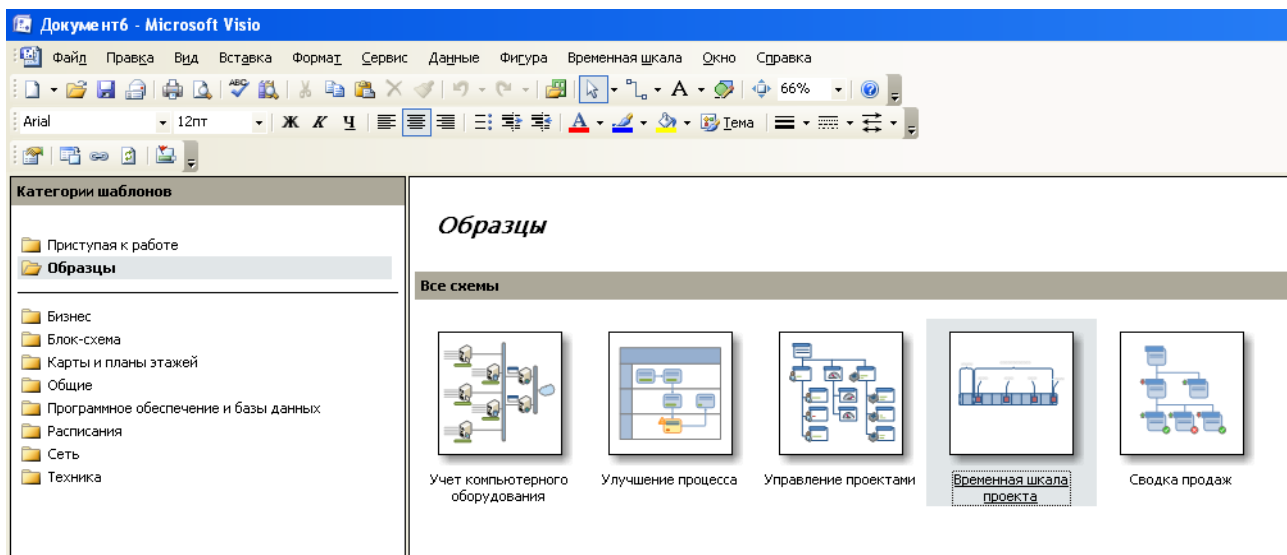
Для цього необхідно виконати наступні дії (Файл> Створити> Розклади> Часова шкала)



Або є можливість проглянути вже готові шаблони, для цього необхідно виконати наступні дії.

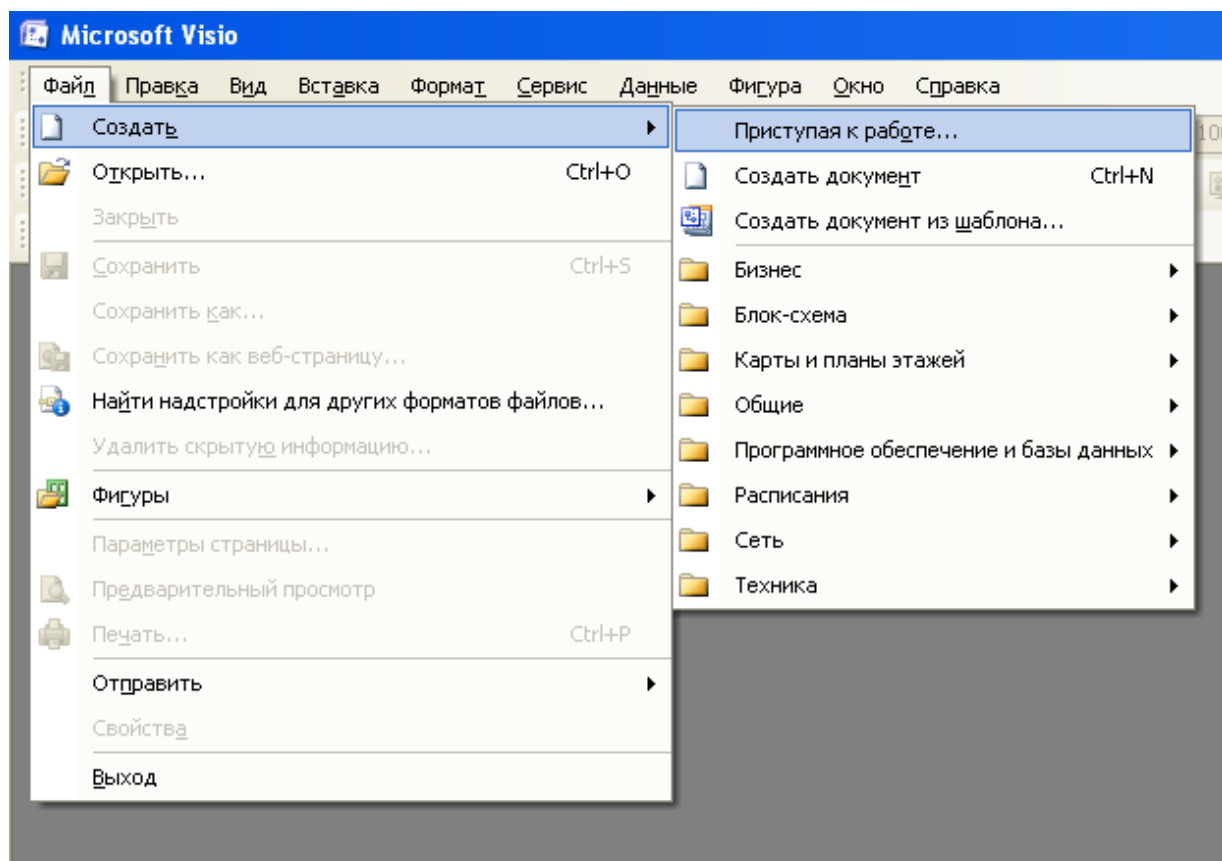


Зайти у закладку «Зразки» й на зразках дійти до шаблону «Часова шкала проекту»



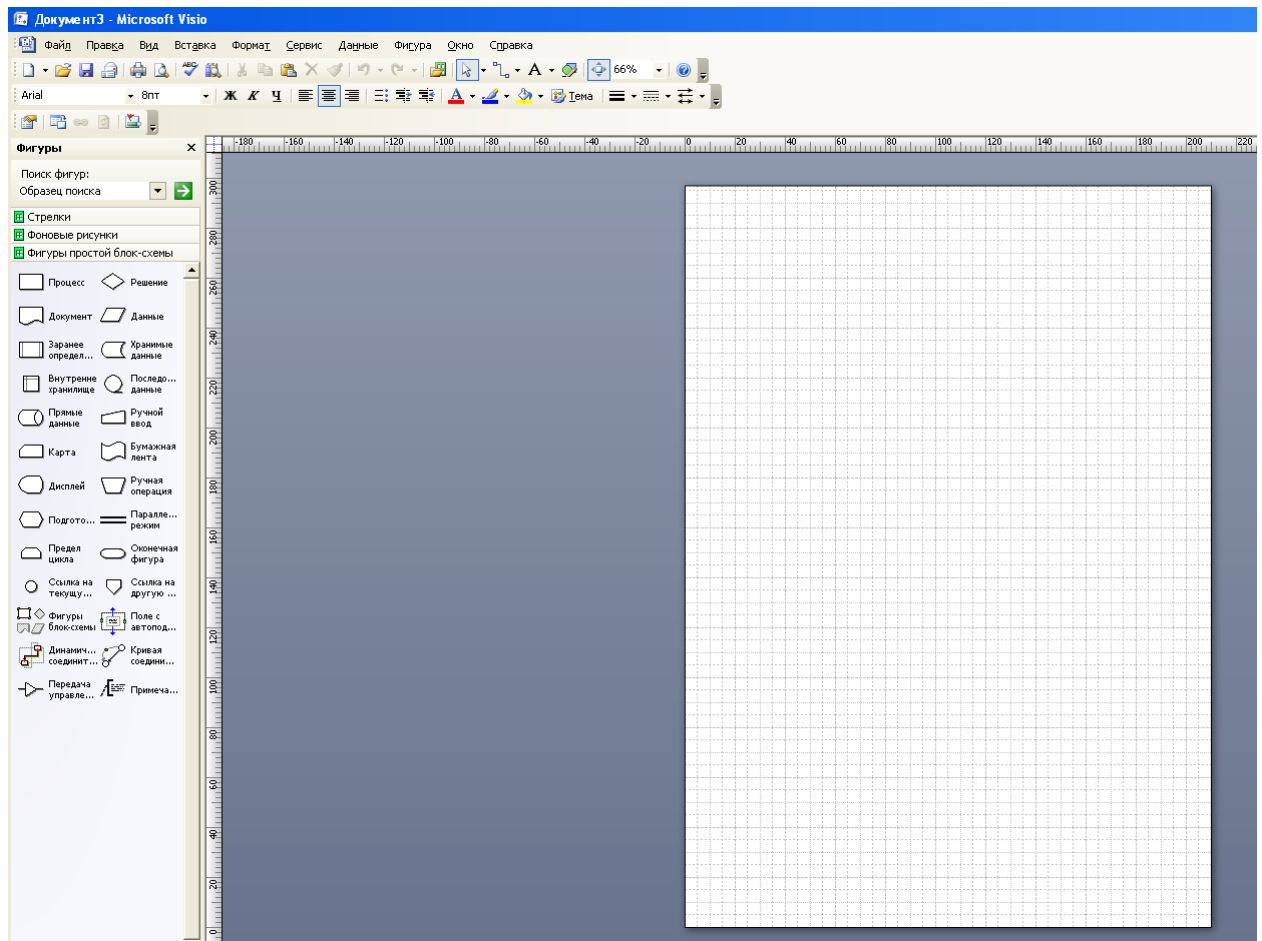
3. Описати структуру проекту розробленого у ході виконання лабораторних робіт № 1-2.

Для цього виконуємо наступні дії: (Файл> Створити> Приступаючи до роботи).



Заходимо у закладку «Бизнес» й обираємо «Проста блок-схема»





Після цього перетягуючи відповідні фігури на полотно створити блок-схему виконання проекту, який був описаний у Лабораторній роботі №1-2.

4. Створити блок схему програмного забезпечення, специфікація якого наведена у лабораторній роботі №3.

Дії аналогічно пункту 3.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Перелік задач та підзадач.

Реалізоване завдання з використанням Visio 2007.

Скріншоти реалізованих завдань.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованих у Visio 2007 завдань.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

### **КОНТРОЛЬНІ ЗАПИТАННЯ:**

1. Навіщо потрібен UML?
2. Що таке CASE система?
3. Що таке Visio 2007?
4. У яких випадках UML буде заважати розробці, а в яких допомагати?
5. Що таке системи реального часу?

## **Лабораторна робота №5 (семестр 5)**

**ТЕМА:** Розробка програми відповідно специфікації.

**МЕТА:** Навчитися розробляти програми відповідно до заданих вимог.

**ЗНАТИ:** Мову програмування C або Pascal.

**ВМІТИ:** Користуватися RAD середовищем (наприклад Builder або Delphi).

**ТЕОРЕТИЧНІ ВІДОМОСТІ.** У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

### **ЗАВДАННЯ:**

На основі ЛР № 3 і доробок у ЛР № 4 – розробити програму, яка повністю відповідає розробленій специфікації.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Перелік задач та підзадач.

Приведена специфікація.

Скріншот реалізованого програмного продукту, який повністю відповідає розробленій специфікації

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованого програмного продукту, який повністю відповідає розробленій специфікації.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

### **КОНТРОЛЬНІ ЗАПИТАННЯ:**

1. Які можуть виникнути труднощі при роботі зі специфікаціями замовника?
2. У яких випадках можна відходити від вимог специфікації?
3. Які існують стандарти для документування і супроводу програми?
4. Що таке сертифікація?
5. Як на вашу думку, реалізувати сертифікацію в розробленій програмі?

## **Лабораторна робота №6 (семестр 5)**

**ТЕМА:** Створення файлу допомоги до розробленої програми.

**МЕТА:** Навчитися складати технічні документи і керівництво для користувача відповідно до ГОСТа на інформаційні технології (РД50-34.698-90).

Отримати навички подання керівництва для користувача в електронному форматі Microsoft Help File (hlp).

**ЗНАТИ:** Мову програмування C або Pascal.

**ВМІТИ:** Інсталювати RAD (наприклад Builder або Delphi). Знати основи розробки програм під ОС Windows.

**ТЕОРЕТИЧНІ ВІДОМОСТІ.** У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

### **ЗАВДАННЯ.**

Скласти керівництво для користувача програми розробленої в ЛР №5 відповідно до ГОСТ на інформаційні технології РД50-34.698-90 і зробити електронну версію цього посібника у форматі довідкової системи MS Help.

Користуватися компілятором електронної довідкової системи MS HelpWorkshop версії 4х або вище. Можливе використання іншого компілятора з узгодженням у лектора.

### **ПРИКЛАД ПОРЯДКУ ВИКОНАННЯ РОБОТИ.**

1. У редакторі MS Word створити керівництво користувача для реалізованого програмного продукту.

2. Виділити в документі основні розділи. Підготувати його до публікації в електронній формі.

3. У компіляторі довідкової системи MS HelpWorkshop створити проект електронного керівництва.

4. Створити файл змісту керівництво користувача в програмі MS HelpWorkshop.

5. Підключити файл змісту в розроблену програму (ЛР № 5), відкомпілювати проект, перевірити створений Help-файл.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Наявність керівництва для користувача програми розробленої в ЛР №5 відповідно до ГОСТ на інформаційні технології РД50-34.698-90

Наявність скріншота реалізованого керівництва користувача у форматі довідкової системи MS Help (\*. hlp);.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність керівництва для користувача програми розробленої в ЛР №5 відповідно до ГОСТ на інформаційні технології РД50-34.698-90

Наявність реалізованого керівництва користувача у форматі довідкової системи MS Help (\*. hlp).

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

**КОНТРОЛЬНІ ЗАПИТАННЯ:**

1. Навіщо необхідна довідкова система в програмі?

2. Які відомості потрібні в довідковій системі, а які ні?

3. На Вашу думку, який оптимальний розмір довідкової системи (обґрунтувати відповідь)?

## Лабораторна робота №7 (семестр 5)

**ТЕМА:** Організація тестування розробленої програми-додатку.

**МЕТА:** Отримати практичні навички в тестуванні програмного забезпечення.

**ЗНАТИ:** Мову програмування C або Pascal.

**ВМІТИ:** Інсталювати RAD (наприклад Builder або Delphi). Знати основи розробки програм під ОС Windows.

### ТЕОРЕТИЧНІ ВІДОМОСТІ.

Теоретичні відомості наведені у лекції № 6.

*Тестуванням* називається процес виконання програми з метою виявлення помилки. Ніяке тестування не може довести відсутність помилок в програмі. Тест – це сукупність вхідних та вихідних даних, отриманих до виконання програми.

*Вихідними даними* для етапу тестування є технічне завдання, специфікація та розроблені на попередніх етапах структурна і функціональна схеми програмного продукту. Під час тестування рекомендується дотримуватися таких основних принципів:

- Передбачувані результати повинні бути відомі до тестування.
- Необхідно досконально вивчати результати кожного тесту.
- Необхідно перевіряти дії програми на невірних даних.
- Необхідно перевіряти програму на несподівані побічні ефекти.
- Вдалим вважається тест, який виявляє хоча б одну ще не виявлену помилку.
- Імовірність наявності помилки в частині програми пропорційна кількості помилок, вже виявлених в цій частині.

*Процес розробки ПЗ* передбачає три стадії тестування:

1. Автономне тестування компонентів ПЗ;
2. Комплексне тестування розробленого ПЗ;
3. Системне або оцінкове тестування.

Існує два підходи до формування тестів:

- Структурний; Тестування за принципом "білого ящика";
- Функціональний; Тестування за принципом «чорної ящика».

### **Тестування за принципом "білого ящика"**

Стратегія тестування за принципом "білого ящика", або стратегія тестування, керована логікою програми (з урахуванням алгоритму), дозволяє перевірити внутрішню структуру програми. У цьому випадку тестує отримує тестові дані шляхом аналізу логіки програми.

### **Тестування на основі потоку управління**

Стратегія «білого ящика» включає в себе наступні критерії структурного тестування:

- C0 – критерій покриття операторів;
- C1 – критерій покриття рішень;
- C2 – критерій покриття усіх шляхів у керуючому графові програми.

### **Тестування на основі потоку даних**

Стратегія «білого ящика» включає в себе наступні критерії структурного тестування:

- CP – покриття пар досяжних дуг у керуючому графові програми.



## Оформлення результатів тестування

Результати тестування за принципом "білого ящика» можна оформити у вигляді таблиці:

Номер тесту	Критерій тестування	Призначення тесту	Значення вихідних даних	Очікуваний результат	Отриманий результат

## Тестування за принципом «чорного ящика»

Одним із способів перевірки програм є стратегія тестування, звана стратегією "чорного ящика" або тестуванням з керуванням за даними. У цьому випадку програма розглядається як "чорний ящик" і таке тестування має на меті з'ясування обставин, в яких поведінка програми не відповідає специфікації.

Стратегія "чорного ящика" включає в себе такі методи формування тестових наборів:

- еквівалентне розбиття;
- аналіз граничних значень;
- аналіз причинно-наслідкових зв'язків;
- припущення про помилку.

## Виділення класів еквівалентності

Класи еквівалентності виділяються шляхом вибору кожної вхідної умови (зазвичай це пропозиція або фраза з специфікації) та розбивкою його на дві або більше груп. Для цього використовується таблиця наступного вигляду:

Вхідна умова	Правильні класи еквівалентності	Неправильні класи еквівалентності

Правильні класи включають правильні дані, неправильні класи – неправильні дані.

Виділення класів еквівалентності є евристичним процесом, проте при цьому існує ряд правил:

1. Якщо вхідні умови описують область значень (наприклад «ціле може приймати значення від 1 до 20»), то виділяють один правильний клас і два неправильних  $X < 1$  і  $X > 20$ .

2. Якщо вхідна умова описує ситуацію "повинно бути" (наприклад, «першим символом ідентифікатора повинна бути буква»), то визначається один правильний клас еквівалентності (перший символ – буква) і один неправильний (перший символ – не буква).

3. Якщо є будь-яка підстава вважати, що різні елементи класу еквівалентності трактуються програмою неоднаково, то даний клас розбивається на менші класи еквівалентності.

### **Побудова тестів**

Цей крок полягає у використанні класів еквівалентності для побудови тестів. Цей процес включає в себе:

- призначення кожному класу еквівалентності унікального номера.
- проектування нових тестів, кожен з яких покриває як можна більше число непокритих класів еквівалентності, до тих пір, поки всі правильні класи не будуть покриті (тільки не загальними) тестами.
- запис тестів, кожен з яких покриває один і тільки один з непокритих неправильних класів еквівалентності, до тих пір, поки всі неправильні класи не будуть покриті тестами.
- розробка індивідуальних тестів для неправильних класів еквівалентності обумовлено тим, що певні перевірки з помилковими входами приховують або замінюють інші перевірки з помилковими входами.

Недоліком методу еквівалентного розбиття є те, що він не досліджує комбінації вхідних умов.

### **Аналіз граничних значень**

Граничні умови – це ситуації, що виникають на, вище або нижче меж вхідних класів еквівалентності. Аналіз граничних значень відрізняється від

еквівалентного роздроблення наступним:

Вибір будь-якого елемента в класі еквівалентності в якості представницького при аналізі граничних умов здійснюється таким чином, щоб перевірити тестом кожную границю цього класу.

При розробці тестів розглядаються не тільки вхідні умови (простір входів), але і простір результатів.

Застосування методу аналізу граничних умов вимагає певної міри творчості і спеціалізації в розглядаємій проблемі. Тим не менш, існує кілька загальних правил цього методу:

1. Побудувати тести для границь області та тести з неправильними вхідними даними для ситуацій незначного виходу за межі області, якщо вхідна умова описує область значень (наприклад, для області вхідних значень від -1.0 до +1.0 необхідно написати тести для ситуацій -1.0, +1.0, -1.001 і +1.001).

2. Побудувати тести для мінімального і максимального значень умов і тести, більше і менше цих двох значень. Якщо вхідна умова задовольняє дискретного ряду значень, наприклад, якщо вхідний файл може містити від 1 до 255 записів, то перевірити 0,1,255 і 256 записів.

3. Якщо вхід або вихід програми є впорядкована множина (наприклад, послідовний файл, лінійний список, таблиця), то зосередити увагу на першому і останньому елементах цієї множини.

### **Припущення про помилку**

Часто програміст з великим досвідом вишукує помилки "без всяких методів". При цьому він підсвідомо використовує метод "припущення про помилку". Процедура методу припущення про помилку в значній мірі заснована на інтуїції. Основна ідея методу полягає в тому, щоб перерахувати в деякому списку можливі помилки або ситуації, в яких вони можуть з'явитися, а потім на основі цього списку скласти тести. Іншими словами, потрібно перерахувати ті спеціальні випадки, які можуть бути не враховані при проектуванні.

## Оформлення результатів тестування

Результати тестування за принципом «чорного ящика» можна оформити у вигляді таблиці:

Номер тесту	Призначення тесту	Значення вихідних даних	Очікуваний результат	Реакція програми	Висновок

## Тестування для користувацького інтерфейсу

При розробці набору тестів для аналізу користувацького інтерфейсу необхідно оцінити наступні характеристики інтерфейсу програми:

### 1. Функціональність:

- правильне виконання базових функцій;
- наявність пропущених функцій;
- робота в реальному часі;
- надмірність або недостатність вихідної інформації;
- способи її збереження і представлення.

### 2. Організація екрану:

- естетичне оформлення екрана;
- наявність меню;
- організація діалогових вікон;
- можливість управляти надлишковою інформацією на екрані.

### 3. Організація меню:

- складність ієрархії меню;
- правила переходу по меню (вверх, вниз, швидкий вибір);
- однозначна відповідність команд і пунктів меню;
- наявність контекстного меню.

### 4. Довідкова система:

- складність і повнота викладу;

- багатослівність і емоційність;
- помилки викладу.

#### 5. Обробка помилок користувача:

- за вхідними даними;
- виконання функцій в неправильному контексті.

### **Оціночне тестування**

Оціночне тестування включає такі види:

- тестування зручність використання;
- тестування на граничних обсягах;
- тестування на граничних навантаженнях;
- тестування захисту та надійності;
- тестування продуктивності при різній апаратурі;
- тестування зручності установки і обслуговування, сумісності;

### **Ручне тестування**

Ручне тестування полягає у виконанні документування процедури, де описана методика виконання тестів, що задає порядок тестів і для кожного тесту список значень параметрів, які подаються на вхід і список результатів, очікуваних на виході. Контроль відповідності результатів очікуваних (тестів) виконує людина.

### **Автоматизоване тестування**

Автоматизоване тестування передбачає створення тестового драйвера, де описана методика виконання тестів, що задає порядок тестів і для кожного тесту список значень параметрів, який подається на вхід і список результатів, очікуваних на виході. Контроль відповідності результатів очікуваних (тестів) повинен виконуватися автоматично (порівняти вміст двох файлів) і видати повідомлення Так / Ні і який тест не пройшов у разі Ні.

## **Особливості тестування для об'єктно-орієнтованих програм**

Основними методами тестування для об'єктно-орієнтованих програм є:

- модульне тестування;
- інтеграційне тестування;
- системне тестування.

**Модульне** тестування – це тестування класів (за принципом білого ящика), використовуючи написання тестових драйверів для перевірки функціональності, що входять в клас методів.

**Інтеграційне** тестування – тестування за принципом білого ящика правильності взаємодії класів, що входять у різні модулі. Взаємодія об'єктів являє собою просто запит одного об'єкта на виконання іншим об'єктом однієї з операцій одержувача і всіх видів обробки, необхідних для завершення цього запиту. Способи взаємодії класів:

- загальнодоступна операція має параметри об'єктного типу;
- загальнодоступна операція повертає значення об'єктного типу;
- метод одного класу створює екземпляр іншого класу як частина соєю реалізації;
- метод одного класу посилається на глобальний екземпляр іншого класу.

**Системне** тестування – тестування за принципом чорної ящика правильності функціонування системи в цілому.

## **ПРИКЛАД ПРОТОКОЛУ ЗА РЕЗУЛЬТАТАМИ ТЕСТУВАННЯ**

### **Звіт про виявлену невідповідність при тестуванні**

**Програма** Калькулятор **Випуск 2** **Дата** XX.XX.XX

**Група розробки:** Інтерфейс

**Тип помилки**

1. **Серйозна**                      Невірне число відображається у правому нижньому куті
  
2. **Незначна**                      Не можу зробити ширину стовпця рівної 17

### **Група розробки: Обчислення**

1. **Фатальна**                      Збій програми, якщо результат обчислення виявляється довше п'яти цифр.
  
2. **Фатальна**                      Нескінченний цикл по таблицях в яких більше 10 рядків.

### **Таблиця проведених тестів**

Номер тесту	Призначення тесту	Значення вихідних даних	Очікуваний результат	Реакція програми	Висновок
1					
2					
3					
4					
...	...	...	...	...	...
N					

**Тестування проводив: П.І.Б.**

## ВИСНОВКИ ЗА РЕЗУЛЬТАТАМИ ТЕСТУВАННЯ

Програма вимагає подальшої доробки для усунення виявлених при тестуванні помилок.

Можливе використання програми, але в наступних версіях потрібно усунути, виявлені при тестуванні помилки.

### **ЗАВДАННЯ:**

Скласти набір тестів для перевірки розробленої програми в ЛР № 6, виконати тести і проаналізувати отримані результати. Ступінь складності та кількість тестів визначається, виходячи з написаних процедур і функцій.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Складені тести, результат тестування, супутній графічний матеріал.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Скласти тести.

Виконати тестування, заповнити протокол результатів тестування.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

### **КОНТРОЛЬНІ ЗАПИТАННЯ:**

Що таке тестування?

Навіщо тестування необхідно?

У яких випадках можна обійтися без тестування?

Що таке модульне тестування?

Що таке інтеграційне тестування?



## **Лабораторна робота №8 (семестр 5)**

**ТЕМА:** Створення інсталяційного пакету розробленої програми.

**МЕТА:** Знайомство із засобом для створення інсталяційного пакета (дистрибутива) розробленої програми.

**ЗНАТИ:** Мову програмування C або Pascal.

**ВМІТИ:** Працювати в ОС Windows.

**ТЕОРЕТИЧНІ ВІДОМОСТІ.** Теоретичні відомості наведені у лекції № 7.

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

**ЗАВДАННЯ:**

Як остаточний результат виконання ЛР № 3 – ЛР № 7.

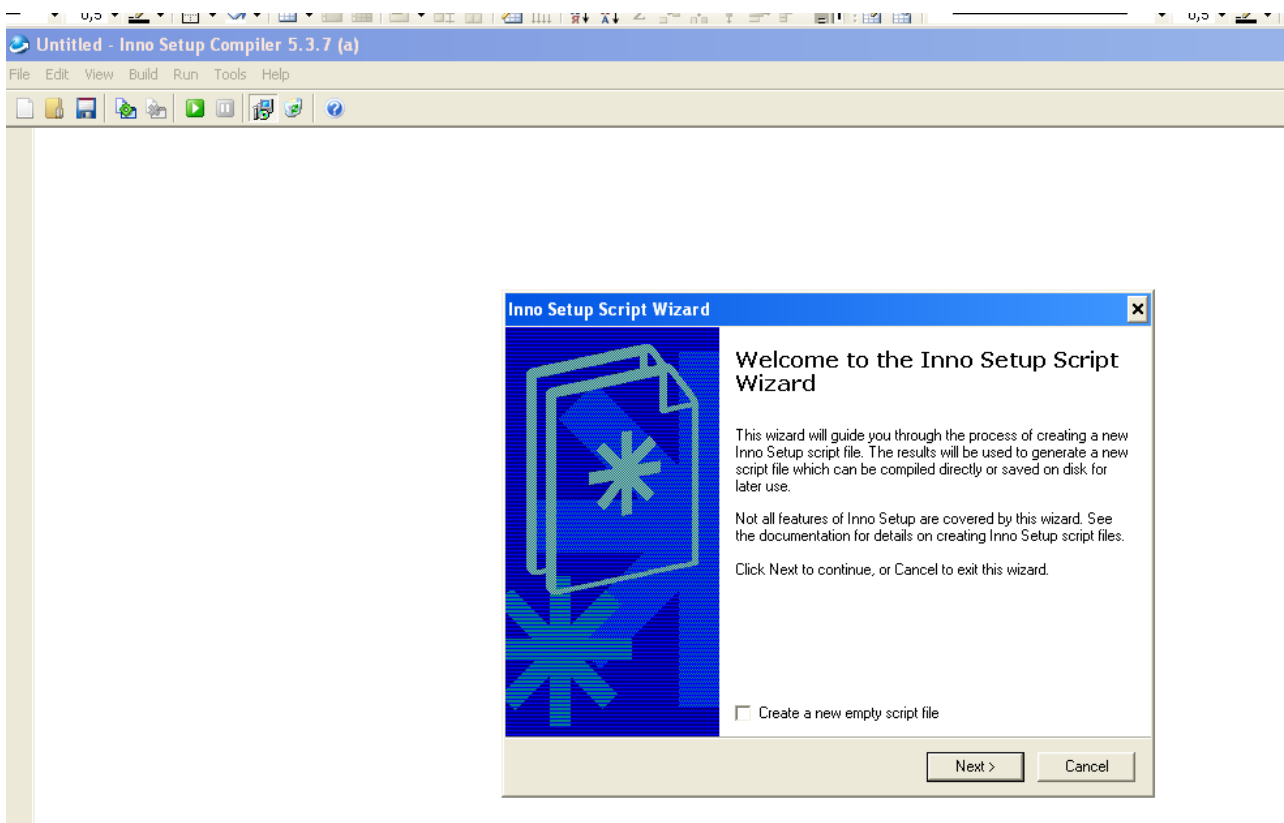
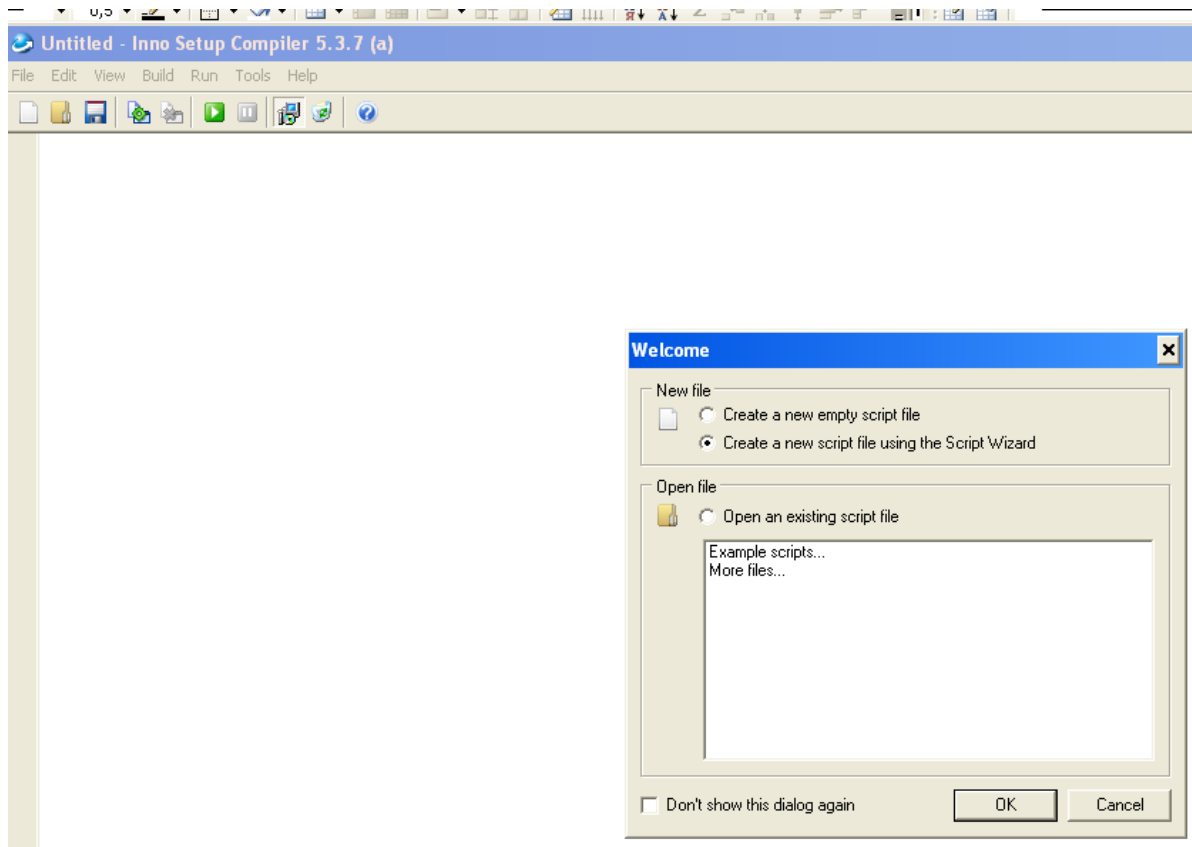
Вибрати ліцензію поширення розробленої програми.

Створити інсталяційний пакет розробленої програми.

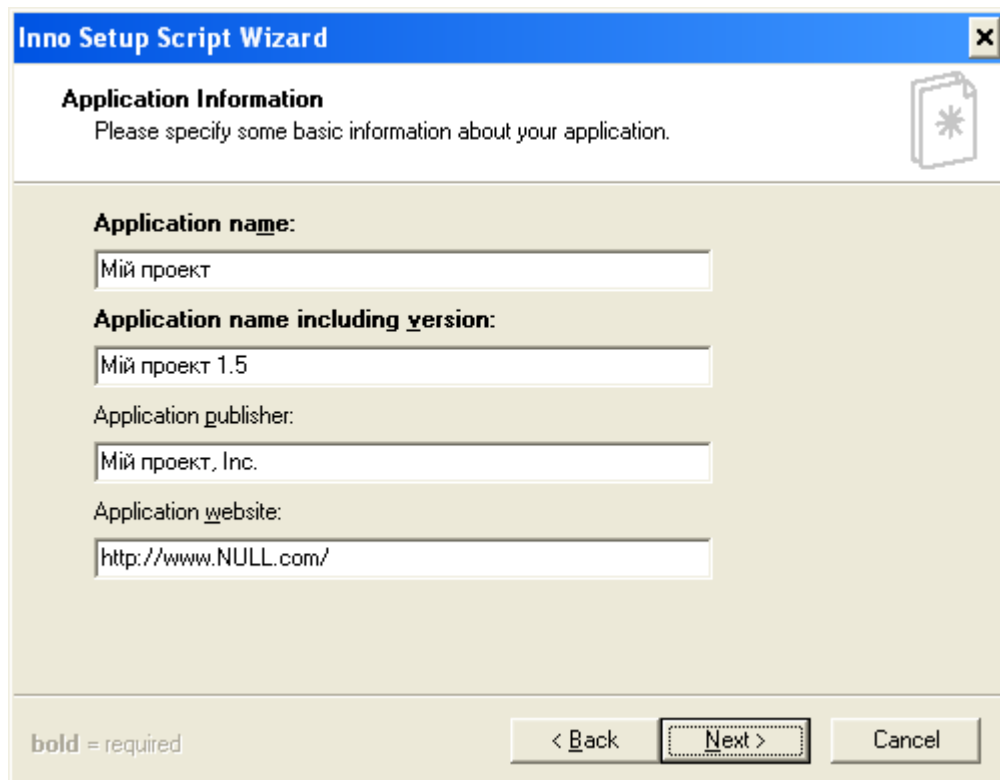
**ХІД ВИКОНАННЯ РОБОТИ**

**1. Запускаємо Inno Setup Compiler.**

При запуску вискакує вікно, у якому обираємо створення нового скрипту інсталяційного пакету («Create a new script file using the Script Wizard»).



## 2. Вводимо інформацію про проект для інсталлятора



The screenshot shows the 'Inno Setup Script Wizard' window, specifically the 'Application Information' step. The window has a blue title bar and a close button. The main area is light beige. At the top, it says 'Application Information' and 'Please specify some basic information about your application.' Below this, there are four text input fields: 'Application name:' (containing 'Мій проект'), 'Application name including version:' (containing 'Мій проект 1.5'), 'Application publisher:' (containing 'Мій проект, Inc.'), and 'Application website:' (containing 'http://www.NULL.com/'). At the bottom, there is a legend 'bold = required', a '< Back' button, a 'Next >' button (which is highlighted with a dashed border), and a 'Cancel' button.

**Application Information**  
Please specify some basic information about your application.

**Application name:**  
Мій проект

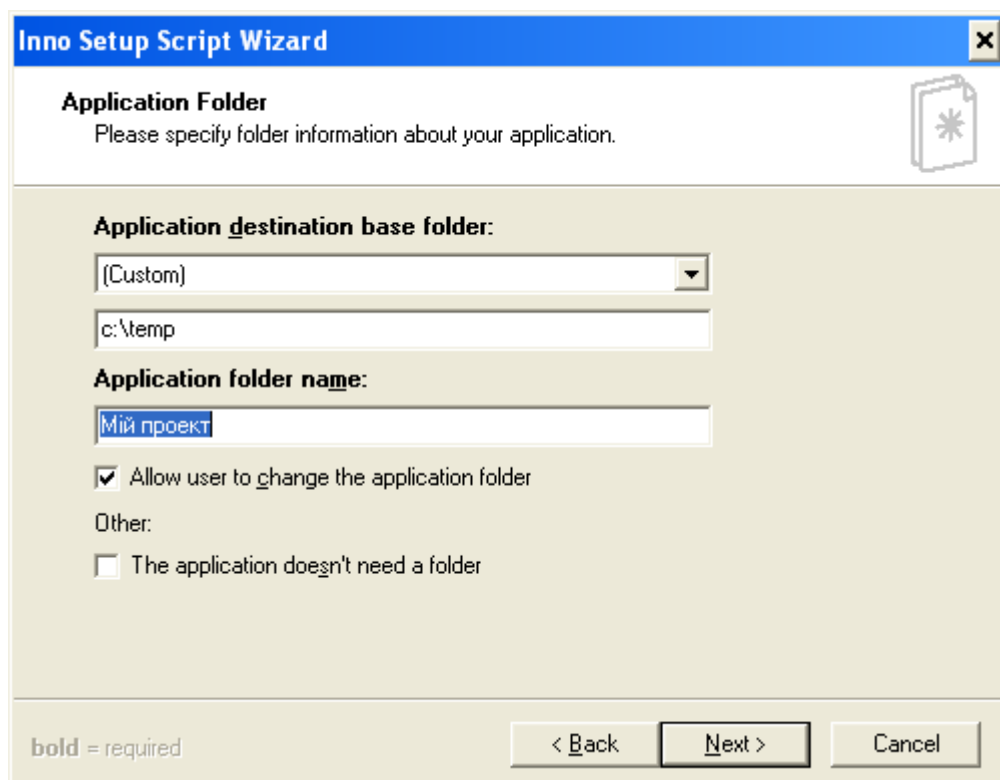
**Application name including version:**  
Мій проект 1.5

Application publisher:  
Мій проект, Inc.

Application website:  
http://www.NULL.com/

bold = required      < Back    Next >    Cancel

## 3. Обираємо шлях куди автоматично буде встановлюватися розроблене студентом програмне забезпечення.



The screenshot shows the 'Inno Setup Script Wizard' window, specifically the 'Application Folder' step. The window has a blue title bar and a close button. The main area is light beige. At the top, it says 'Application Folder' and 'Please specify folder information about your application.' Below this, there are two text input fields: 'Application destination base folder:' (containing '(Custom)') and 'c:\temp'. Below these, there is a text input field for 'Application folder name:' (containing 'Мій проект'). There are two checkboxes: 'Allow user to change the application folder' (checked) and 'The application doesn't need a folder' (unchecked). At the bottom, there is a legend 'bold = required', a '< Back' button, a 'Next >' button (which is highlighted with a dashed border), and a 'Cancel' button.

**Application Folder**  
Please specify folder information about your application.

**Application destination base folder:**  
(Custom)  
c:\temp

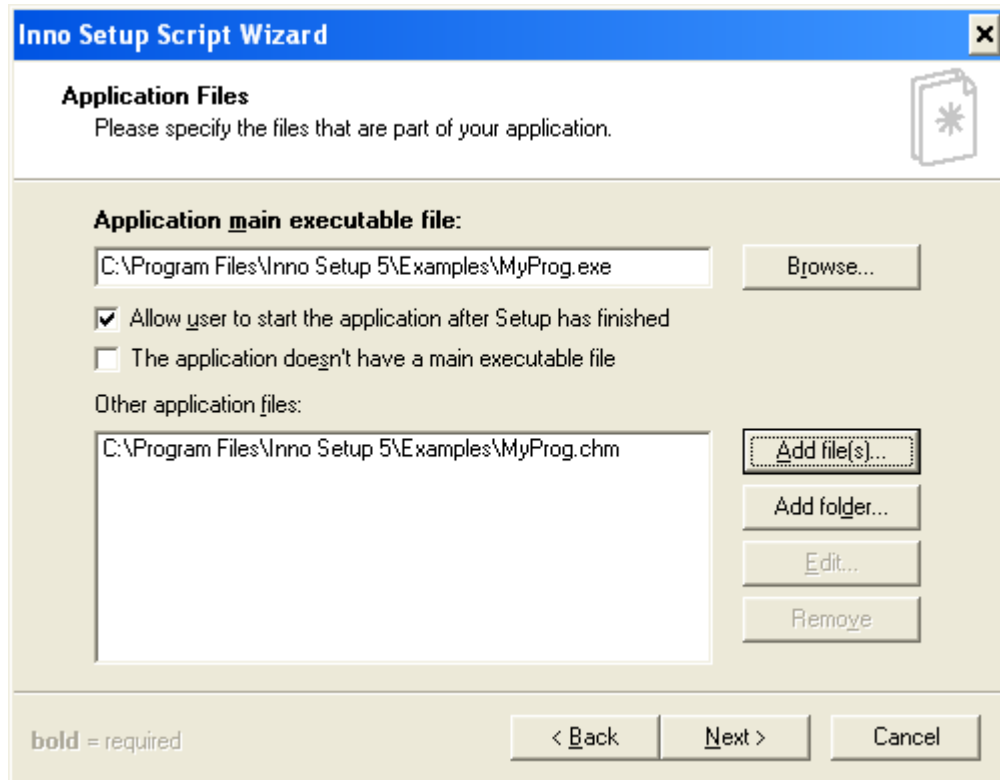
**Application folder name:**  
Мій проект

☒ Allow user to change the application folder

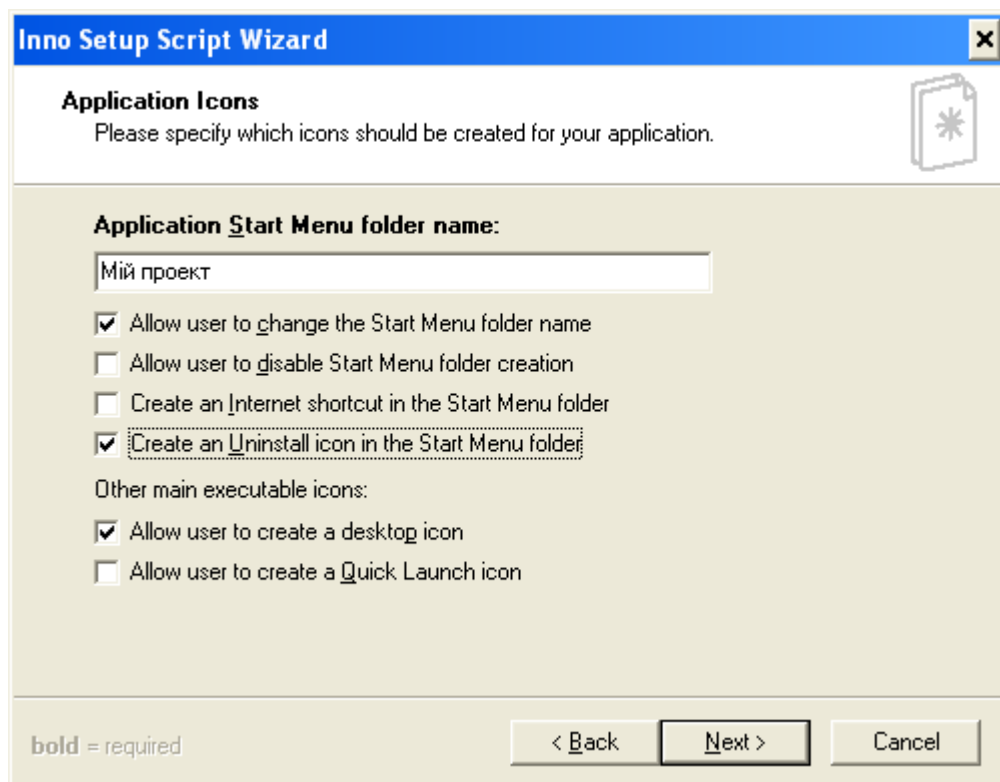
Other:  
☐ The application doesn't need a folder

bold = required      < Back    Next >    Cancel

4. Обираємо шлях де знаходитися розроблене студентом програмне забезпечення, яке буде інсталиюватися.



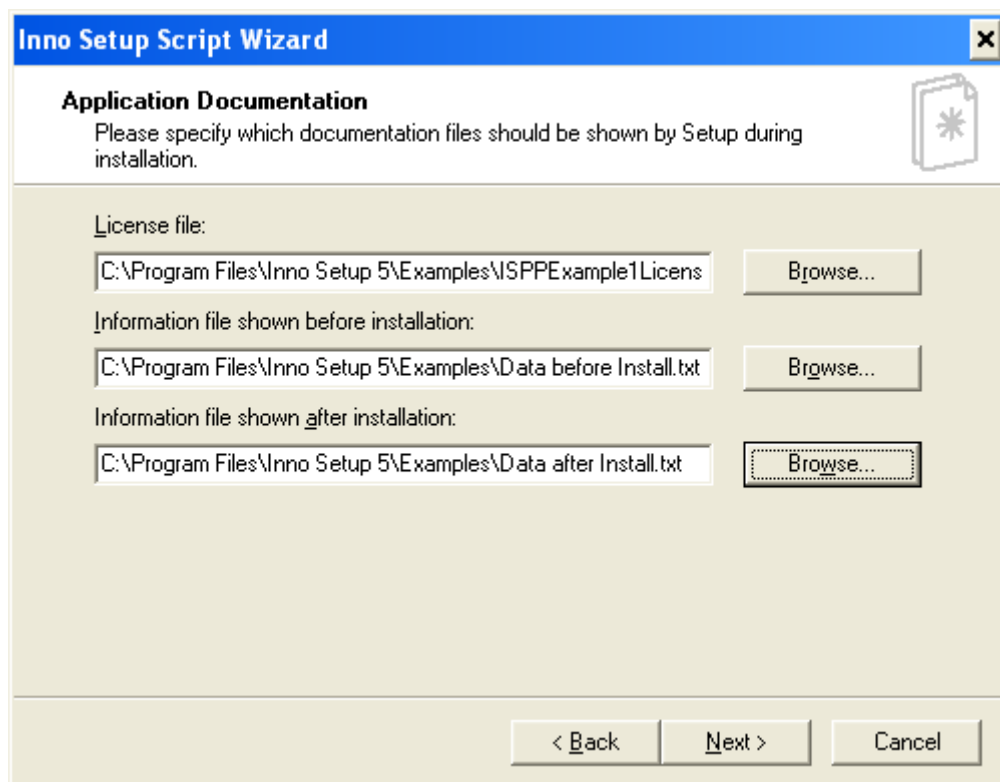
5. Вводимо назву проекту, яка буде відображатися у  
Пуск → Програми → ...



При цьому вказуємо чи створювати одночасно й деінсталятор у цьому меню.

## **6. Встановлюємо шляхи до наступних файлів:**

- файлу ліцензії;
- файлу у якому міститься інформація до інсталяції;
- файлу у якому міститься інформація після інсталяції.



## 7. Обираємо мову інсталяції.



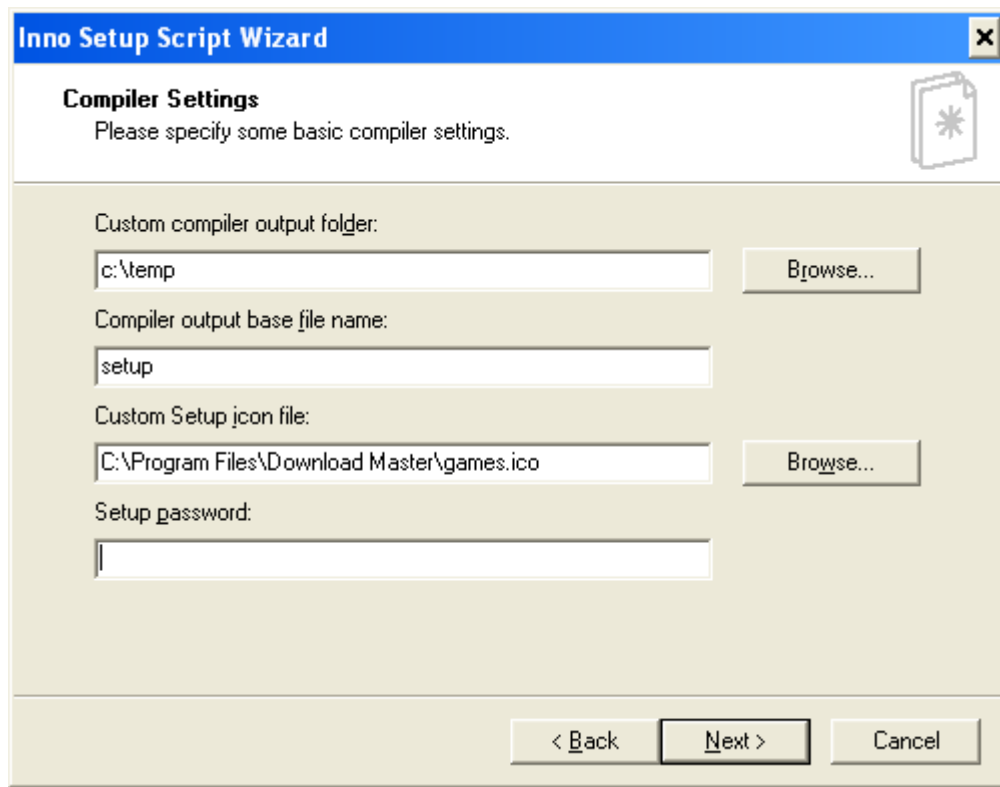
## 8. Встановлюємо наступні параметри:

– Шлях до папки у якій буде зберігатися скомпільований файл інсталятора.

Ім'я інсталяційного файлу.

Іконка інсталяційного файлу.

Пароль доступу до інсталяційного файлу.

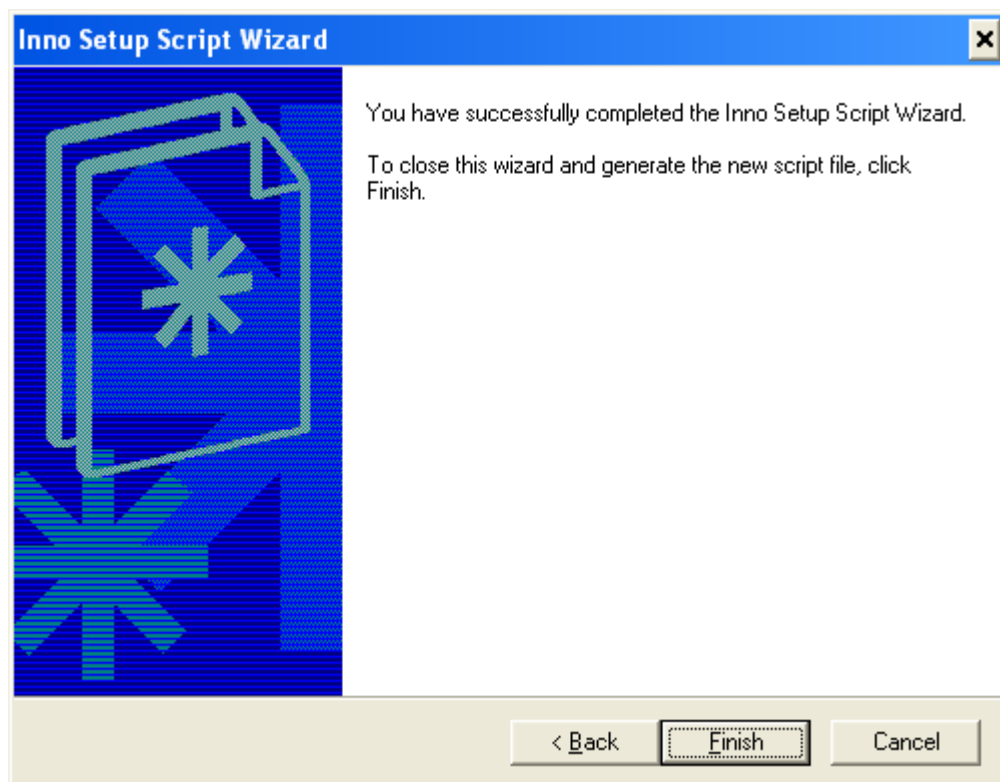


The screenshot shows the 'Compiler Settings' window of the Inno Setup Script Wizard. The title bar reads 'Inno Setup Script Wizard'. The main heading is 'Compiler Settings' with a sub-instruction: 'Please specify some basic compiler settings.' Below this, there are four input fields with corresponding 'Browse...' buttons:

- 'Custom compiler output folder:' with the text 'c:\temp' and a 'Browse...' button.
- 'Compiler output base file name:' with the text 'setup'.
- 'Custom Setup icon file:' with the text 'C:\Program Files\Download Master\games.ico' and a 'Browse...' button.
- 'Setup password:' with an empty text field.

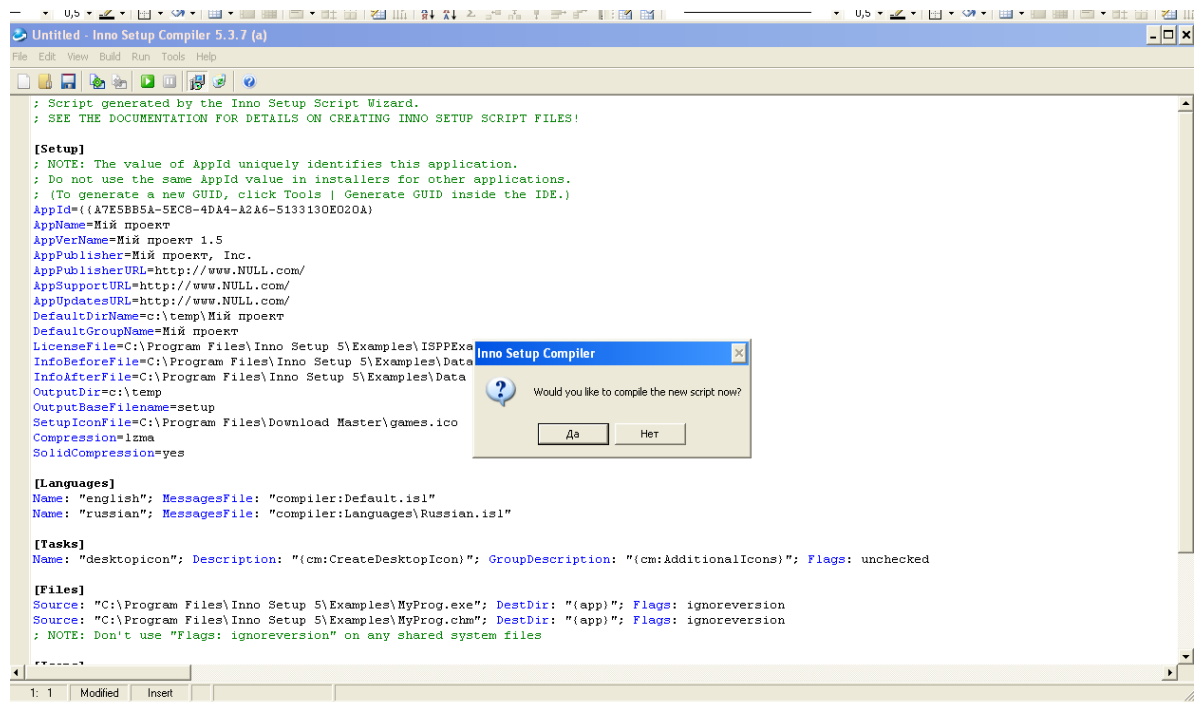
At the bottom of the window are three buttons: '< Back', 'Next >', and 'Cancel'.

## 9. Закінчуємо роботу з Inno Setup Script Wizard.



The screenshot shows the completion screen of the Inno Setup Script Wizard. The title bar reads 'Inno Setup Script Wizard'. On the left is a large graphic of a document with a star. The text on the right says: 'You have successfully completed the Inno Setup Script Wizard. To close this wizard and generate the new script file, click Finish.' At the bottom are three buttons: '< Back', 'Finish', and 'Cancel'.

**10. Виводиться вікно, у якому знаходиться автоматично згенерований скрипт інсталяційного пакету, та віконце запиту на компіляцію.**



**11. Виводиться вікно з результатами створення інсталяційного пакету.**



111 - Inno Setup Compiler 5.3.7 (a)
File Edit View Build Run Tools Help

```

; Script generated by the Inno Setup Script Wizard.
; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

[Setup]
; NOTE: The value of AppId uniquely identifies this application.
; Do not use the same AppId value in installers for other applications.
; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)
AppId={{A7E5BB5A-5EC8-4DA4-A2A6-5133130E020A}
AppName=Мій проєкт

```

```

**** Starting compile. [15:56:43]

Parsing [Setup] section, line 8
Parsing [Setup] section, line 9
Parsing [Setup] section, line 10
Parsing [Setup] section, line 11
Parsing [Setup] section, line 12
Parsing [Setup] section, line 13
Parsing [Setup] section, line 14
Parsing [Setup] section, line 15
Parsing [Setup] section, line 16
Parsing [Setup] section, line 17
Parsing [Setup] section, line 18
Parsing [Setup] section, line 19
Parsing [Setup] section, line 20
Parsing [Setup] section, line 21
Parsing [Setup] section, line 22
Parsing [Setup] section, line 23
Parsing [Setup] section, line 24
Reading file (LicenseFile)
Reading file (InfoBeforeFile)
Reading file (InfoAfterFile)
Reading file (WizardImageFile)
  File: C:\Program Files\Inno Setup 5\WIZMODERNIMAGE.BMP
Reading file (WizardSmallImageFile)
  File: C:\Program Files\Inno Setup 5\WIZMODERNSMALLIMAGE.BMP
Preparing Setup program executable
  Updating icons (SETUP.EXE)
Reading default messages from Default.isl
Parsing [Languages] section, line 27
  File: C:\Program Files\Inno Setup 5\Default.isl
Parsing [Languages] section, line 28
  File: C:\Program Files\Inno Setup 5\Languages\Russian.isl
Parsing [LangOptions], [Messages], and [CustomMessages] sections
  Messages in script file
Reading [Code] section
Parsing [Tasks] section, line 31
Parsing [Icons] section, line 39
Parsing [Icons] section, line 40
Parsing [Icons] section, line 41
Parsing [Run] section, line 44
Parsing [Files] section, line 34
Parsing [Files] section, line 35
Creating setup files
  Updating icons (SETUP.EXE)
  Compressing: C:\Program Files\Inno Setup 5\Examples\MyProg.exe
  Compressing: C:\Program Files\Inno Setup 5\Examples\MyProg.chm
  Compressing Setup program executable
  Updating version info

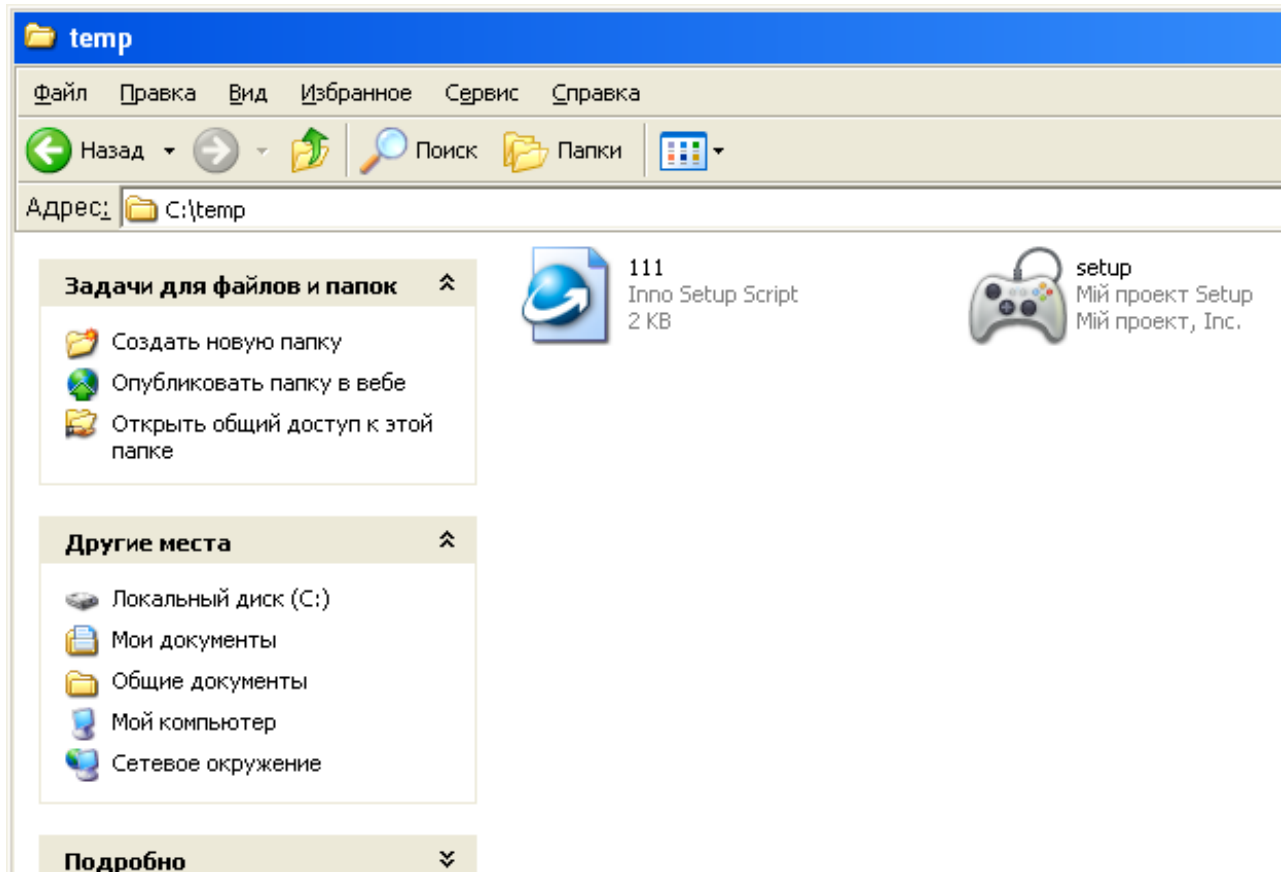
**** Finished. [15:56:45, 00:02.578 elapsed]

```

Compiler Output
Debug Output

1: 1
Insert

**12. У заданій при формуванні інсталяційного пакету папці знаходиться файл інсталяції програмного продукту розробленого студентом.**



На цьому робота вважається виконаною.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Текст інсталяційного скрипту.

Скріншот реалізованих завдань.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованих поставлених завдань.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

**КОНТРОЛЬНІ ЗАПИТАННЯ:**

1. Які існують ліцензії розповсюдження програм?
2. Що таке інсталяційний пакет (дистрибутив)?
3. Навіщо потрібен інсталяційний пакет?
4. У яких випадках інсталяційний пакет потрібен?

## Лабораторна робота № 9 (семестр 6)

**ТЕМА:** Створення власного пакета інсталяції програмного забезпечення

**ЦІЛЬ:** Розширене вивчення процесу інсталяції програмного забезпечення.

**ЗНАТИ:** Мову програмування C або Pascal відповідно до обраної RAD. Основи адміністрування ОС Windows.

**ТЕОРЕТИЧНІ ВІДОМОСТІ:** Теоретичні відомості наведені у лекції № 7.

1. WWW.GOOGLE.UA;
2. WWW.BING.COM;
3. UK.WIKIPEDIA.ORG;
4. RU.WIKIPEDIA.ORG.

### ЗАВДАННЯ.

Необхідно створити власний інсталяційний пакет. Специфікація інсталяційного пакета наступна.

Створювати інсталяційний пакет необхідно для програми, яка була розроблена у ході виконання ЛР № 3-7.

Інсталяційний пакет створюється НЕ АВТОМАТИЧНО як у ЛР №8, а студентом самостійно.

Інсталяційний пакет може бути представлений як один файл (Setup.exe) та папкою з файлами, яка знаходиться окремо від Setup.exe.

Тобто після запуску інсталяційного файлу (Setup.exe)), який вже створений студентом (**усі необхідні файли для інсталяції знаходяться у відповідній папці, не в тій де знаходиться Setup.exe**), виконується установка ПЗ.

## ХІД ВИКОНАННЯ РОБОТИ

Студентом створюється програма (Setup.exe), в якій реалізовано ряд вікон, перехід з одного до іншого відбувається за натисканням кнопки «Далі», як у класичних програмах інсталяції. Кожне вікно (а їх 7) представляє з себе форму, у якій знаходяться наступні дані:

**1 Форма:** Вітання; Назва програми; Версія програми; Назва розробника; Дата інсталяційної зборки.

Якщо на ПК не встановлена ОС Windows XP SP3 (перевіряється за допомогою засобів мови на якій пишеться інсталятор) вивести на екран відповідне повідомлення й завершити роботу інсталятора.

Перевірити чи відбувалася інсталяція даного пакета на ПК (перевірити ключ реєстру див. форма 3, якщо ключ вже прописаний то програма встановлювалася, якщо не прописаний, то програма встановлюється вперше), якщо так вивести на екран відповідне повідомлення з можливістю зупинити інсталяцію програми.

**2 форма:** Спосіб поширення програми (на вибір студента одна з платних ліцензій).

**3 форма:** Уведення серійного номера програми (XXXX-XXXX-XXXX). При правильному введенні S/N продовжити роботу інсталятора.

**Дія 1** Створити гілку реєстру (за допомогою команд мови програмування, на якій пишеться інсталяційний пакет)

HKEY\_LOCAL\_MACHINE\SOFTWARE\KNTU\NAME\

Створити ключі реєстру

Installtime – Час встановлення ПЗ (XX:XX:XX формат 24 часовий

Installdate – Дата встановлення ПЗ (XX:XX:20XX)

Installsn – Серійний номер ПЗ

**Дія 2** Створити текстовий файл Hidekey.dat.

Шлях до файлу “C:\Temp\Hidekey.dat”

**Дія 3** Скритна перевірка з'єднання з Інтернетом (за допомогою команд мови програмування, на якій пишеться інсталяційний пакет) і при його наявності відправити електронний лист по заданій адресі (на вибір студента) з інформацією про встановлення ПЗ.

**4 форма:** Вибір каталогу встановлення програми.

**5 форма:** Налаштування програми (при необхідності).

**6 форма:** Інсталяція програми (створюється відповідний каталог, куди копіюється з заданого каталогу, виконуючий файл програми, й необхідні файли: допомоги й т.і.).

**7 форма:** Вікно завершення роботи інсталятора.

**Після закінчення роботи інсталятора повинні бути виконані наступні пункти:**

Прописаний дані у реєстрі.

Створений каталог, де знаходиться виконуючий файл.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Скріншот реалізованих завдань.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованих поставлених завдань.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

## **КОНТРОЛЬНІ ПИТАННЯ**

1. Що таке подвійне, потрійне й мульти ліцензування?
2. Що таке й навіщо необхідний реєстр?
3. Що таке гілка, ключ реєстру?
4. Що таке Proprietary software?
5. Обмеження на комерційне використання (проприетарного ПЗ)?
6. Обмеження на поширення (проприетарного ПЗ)?
7. Обмеження на модифікацію (проприетарного ПЗ)?
8. Що таке Free Software Foundation (скорочено FSF)?

## **Лабораторна робота №10 (семестр 6)**

**ТЕМА:** Створення власного пакета деінсталяції програмного забезпечення

**ЦІЛЬ:** Розширене вивчення процесу деінсталяції програмного забезпечення.

**ЗНАТИ:** Мова програмування C або Pascal відповідно до обраної RAD. Основи адміністрування ОС Windows.

**ТЕОРЕТИЧНІ ВІДОМОСТІ:** Теоретичні відомості наведені у лекції № 7.

1. WWW.GOOGLE.UA;
2. WWW.BING.COM;
3. UK.WIKIPEDIA.ORG;
4. RU.WIKIPEDIA.ORG.

### **ЗАВДАННЯ.**

Створити програмно файл деінсталяції (Uninstall.exe), який деінсталює програму, встановлену у результаті виконання ЛР № 9 (семестр 6):

Видаляються усі виконуючі файли та папки, де вони були записані.

Залишаються ключі реєстру, вилучається файл “C:\Temp\Hidekey.dat”.

### **ХІД ВИКОНАННЯ РОБОТИ**

Специфікація деінсталяційного пакета.

- 1) Програма – відповідно до ЛР №9 (семестр 6).
- 2) Послідовність деінсталяції.

Програма деінсталяції складається з покрокового виконання наступних



форм.

**1 Форма:** Вітання.

Перевірити встановлені компоненти програми та при їхній наявності продовжити програму.

При натисканні кнопки «Далі», вивести вікно попередження про наслідки видалення інформації із ПК користувача, з можливістю скасування дії.

**2 форма:** Проведення деінсталяції програми з покроковим виведенням дій на форму програми.

Залишити ключі реєстру, вилучити файл “C:\Temp\Hidekey.dat”.

Скритно перевірити з'єднання з Інтернетом і при його наявності відправити електронний лист по заданій адресі (на вибір студента) з інформацією про видалення ПЗ.

**3 форма:** Вікно завершення роботи деінсталяційного пакета.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Скріншот реалізованих завдань.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованих поставлених завдань.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

## **КОНТРОЛЬНІ ПИТАННЯ**

1. Що таке система керування пакетами (у різних ОС)?
2. Що таке репозиторій, приведіть приклади (у різних ОС)?
3. Що таке Installshield?
4. Що таке WISE Installer?
5. Що таке Inno Setup?
6. Для чого потрібні файли формату \*.iss?
7. Для чого потрібні файли формату \*.msi?
8. Фізична структура пакета \*.msi?

## **Лабораторна робота №11 (семестр 6)**

**ТЕМА:** Використання програм моніторингу жорсткого диска й системного реєстру для перевірки інсталяційних пакетів і програм

**ЦІЛЬ:** Вивчення основ моніторингу ОС Windows.

**ЗНАТИ:** Мова програмування С або Pascal відповідно до обраної RAD.  
Основи адміністрування ОС Windows.

### **ТЕОРЕТИЧНІ ВІДОМОСТІ:**

1. WWW.GOOGLE.UA;
2. WWW.BING.COM;
3. <http://ru.wikipedia.org/wiki/Regmon>
4. <http://ru.wikipedia.org/wiki/Filemon>
5. [http://ru.wikipedia.org/wiki/Process\\_Monitor](http://ru.wikipedia.org/wiki/Process_Monitor)
6. <http://ru.wikipedia.org/wiki/Sysinternals>

### **ЗАВДАННЯ.**

На базі програми ЛР №9 провести моніторинг інсталяції програми за допомогою системних утиліт (на вибір студента).

З'ясувати які дії проводить програма інсталяції. Продемонструвати результат моніторингу викладачу.


### **ХІД ВИКОНАННЯ РОБОТИ**


#### **I. Проведемо моніторинг інсталяції програми за допомогою Filemon.**

Для цього виконаємо наступні дії:

1. Запустимо Filemon.

#	Time	Process	Request	Path	Result	Other
973	17:03:49	chrome.exe	IRP_MJ_FLUSH	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage	SUCCESS	
974	17:03:49	chrome.exe	IRP_MJ_WRITE*	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage	SUCCESS	Offset: 0 Length: 4096
975	17:03:49	chrome.exe	IRP_MJ_WRITE*	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage	SUCCESS	Offset: 45056 Length: 4096
976	17:03:49	chrome.exe	IRP_MJ_WRITE*	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage	SUCCESS	Offset: 81920 Length: 4096
977	17:03:49	chrome.exe	IRP_MJ_WRITE*	C:\\$logFile	SUCCESS	Offset: 12288 Length: 4096
978	17:03:49	chrome.exe	IRP_MJ_CLEANUP	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
979	17:03:49	chrome.exe	IRP_MJ_CLOSE	C:\DASD	SUCCESS	
980	17:03:49	chrome.exe	IRP_MJ_CREATE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Attributes: Any Options: Open
981	17:03:49	chrome.exe	IRP_MJ_QUERY_INF...	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileInternalInformation
982	17:03:49	System	IRP_MJ_CLEANUP	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
983	17:03:49	System	IRP_MJ_CREATE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Attributes: N Options: Open
984	17:03:49	ekrn.exe	IRP_MJ_CREATE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileFsVolumeInformation
985	17:03:49	ekrn.exe	IRP_MJ_QUERY_VOL...	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileInternalInformation
986	17:03:49	ekrn.exe	IRP_MJ_QUERY_INF...	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Attributes: A
987	17:03:49	ekrn.exe	FASTIO_QUERY_BASL...	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Size: 3608
988	17:03:49	ekrn.exe	IRP_MJ_CLEANUP	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
989	17:03:49	ekrn.exe	IRP_MJ_CLOSE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Attributes: N Options: Open
990	17:03:49	ekrn.exe	IRP_MJ_CREATE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileFsVolumeInformation
991	17:03:49	ekrn.exe	IRP_MJ_CREATE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileInternalInformation
992	17:03:49	ekrn.exe	IRP_MJ_QUERY_VOL...	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Attributes: A
993	17:03:49	ekrn.exe	IRP_MJ_READ	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Offset: 0 Length: 3608
994	17:03:49	ekrn.exe	IRP_MJ_CLEANUP	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
995	17:03:49	ekrn.exe	IRP_MJ_CLOSE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Attributes: Any Options: Open
996	17:03:49	chrome.exe	IRP_MJ_CREATE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileInternalInformation
997	17:03:49	chrome.exe	IRP_MJ_CREATE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Attributes: Any Options: Open
998	17:03:49	chrome.exe	IRP_MJ_CREATE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Attributes: Any Options: Open
999	17:03:49	chrome.exe	IRP_MJ_QUERY_INF...	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileInternalInformation
1000	17:03:49	System	IRP_MJ_CLEANUP	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
1001	17:03:49	System	IRP_MJ_CLOSE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileObjectInformation
1002	17:03:49	chrome.exe	IRP_MJ_QUERY_INF...	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	FileDispositionInformation
1003	17:03:49	chrome.exe	IRP_MJ_SET_INFORM...	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
1004	17:03:49	chrome.exe	IRP_MJ_CLEANUP	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
1005	17:03:49	chrome.exe	IRP_MJ_CLOSE	C:\DASD	SUCCESS	
1006	17:03:49	chrome.exe	IRP_MJ_CLOSE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
1007	17:03:49	chrome.exe	FASTIO_QUERY_OPEN	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
1008	17:03:49	chrome.exe	FASTIO_LOCK	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Offset: 1073741826 Length: 510
1009	17:03:49	chrome.exe	FASTIO_LOCK	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Excl: No Offset: 1073741826 Leng...
1010	17:03:49	chrome.exe	FASTIO_UNLOCK	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Offset: 1073741825 Length: 1
1011	17:03:49	chrome.exe	FASTIO_UNLOCK	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Offset: 1073741824 Length: 1
1012	17:03:49	chrome.exe	FASTIO_UNLOCK	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	Offset: 1073741826 Length: 510
1013	17:03:49	System	IRP_MJ_CLOSE	C:\Documents and Settings\assa\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\http_vk.com_0_localstorage-journal	SUCCESS	
1014	17:03:49	System	IRP_MJ_WRITE*	C:\WINDOWS\system32	SUCCESS	Offset: 368640 Length: 4096
1015	17:03:50	ekrn.exe	FASTIO_QUERY_OPEN	C:\WINDOWS\system32\Mscf.dll	SUCCESS	
1016	17:03:50	ekrn.exe	IRP_MJ_CREATE	C:\WINDOWS\system32\Mscf.dll	SUCCESS	Attributes: Any Options: Open
1017	17:03:50	ekrn.exe	FASTIO_QUERY_STA...	C:\WINDOWS\system32\Mscf.dll	SUCCESS	Size: 294400
1018	17:03:50	ekrn.exe	IRP_MJ_CLEANUP	C:\WINDOWS\system32\Mscf.dll	SUCCESS	
1019	17:03:50	ekrn.exe	IRP_MJ_CLOSE	C:\WINDOWS\system32\Mscf.dll	SUCCESS	
1020	17:03:50	System	IRP_MJ_WRITE*	C:\WINDOWS	SUCCESS	Offset: 8192 Length: 4096
1021	17:03:51	System	IRP_MJ_WRITE*	C:\\$logFile	SUCCESS	Offset: 2390840 Length: 24576
1022	17:03:51	System	IRP_MJ_WRITE*	C:\\$logFile	SUCCESS	Offset: 8192 Length: 4096
1023	17:03:51	System	IRP_MJ_WRITE*	C:\\$logFile	SUCCESS	Offset: 4096 Length: 4096

2. Натискаємо на  для того, щоб зробити захват поточного стану процесів підсистеми файлів.

3. Натискаємо на  для того, щоб очистити список процесів.

4. Натискаємо кнопку фільтру  де задаємо той процес, який ми хочемо відстежити. У нашому випадку відстежуємо програму інсталяції (Setup.exe), яка створювалася у ЛР № 9.

Filemon Filter

Process Include(s):

Apply

Process Exclude(s):

Reset

Path Include(s):

Cancel


Path Exclude(s):

History Depth:

☒ Reads
 ☒ Writes

Натискаємо «Apply».



5. Натискаємо на  для того, щоб зробити захват поточного стану процесу Setup.exe

6. Запускаємо програму Setup.exe, й дивимось які дії відбуваються при її запуску, та під час її виконання.

#	Time	Process	Request	Path	Result	Other
6544	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\	SUCCESS	Attributes: Any Options: Open Dire...
6545	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\assa\	SUCCESS	FileBothDirectoryInformation
6546	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\	SUCCESS	
6547	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\assa\	SUCCESS	
6548	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\	SUCCESS	Attributes: Any Options: Open Dire...
6549	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\assa\LOCALS~1\	SUCCESS	FileBothDirectoryInformation
6550	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\LOCALS~1\	SUCCESS	
6551	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\assa\LOCALS~1\	SUCCESS	
6552	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\	SUCCESS	Attributes: Any Options: Open Dire...
6553	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\assa\LOCALS~1\Temp\	SUCCESS	FileBothDirectoryInformation
6554	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\LOCALS~1\Temp\	SUCCESS	
6555	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\assa\LOCALS~1\Temp\	SUCCESS	
6556	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\	SUCCESS	Attributes: Any Options: Open Dire...
6557	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\	SUCCESS	FileBothDirectoryInformation
6558	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\	SUCCESS	
6559	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\	SUCCESS	
6560	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS	
6561	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\WINDOWS\AppPatch\sysmain.sdb	SUCCESS	
6562	17:16:25	setup.exe	IRP_MJ_QUERY_INF...	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp	SUCCESS	FileNameInformation
6563	17:16:25	setup.exe	FASTIO_QUERY_OPEN	C:\WINDOWS\WindowsShell.Manifest	SUCCESS	
6564	17:16:25	setup.exe	IRP_MJ_CREATE	C:\	SUCCESS	Attributes: Any Options: Open Dire...
6565	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\	SUCCESS	FileBothDirectoryInformation
6566	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\	SUCCESS	
6567	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\	SUCCESS	
6568	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\	SUCCESS	Attributes: Any Options: Open Dire...
6569	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\	SUCCESS	FileBothDirectoryInformation
6570	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\	SUCCESS	
6571	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\	SUCCESS	
6572	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\	SUCCESS	Attributes: Any Options: Open Dire...
6573	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\assa\	SUCCESS	FileBothDirectoryInformation
6574	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\	SUCCESS	
6575	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\assa\	SUCCESS	
6576	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\	SUCCESS	Attributes: Any Options: Open Dire...
6577	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\assa\LOCALS~1\	SUCCESS	FileBothDirectoryInformation
6578	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\LOCALS~1\	SUCCESS	
6579	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\assa\LOCALS~1\	SUCCESS	
6580	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\	SUCCESS	Attributes: Any Options: Open Dire...
6581	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\assa\LOCALS~1\Temp\	SUCCESS	FileBothDirectoryInformation
6582	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\LOCALS~1\Temp\	SUCCESS	
6583	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\assa\LOCALS~1\Temp\	SUCCESS	
6584	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\	SUCCESS	Attributes: Any Options: Open Dire...
6585	17:16:25	setup.exe	IRP_MJ_DIRECTORY_...	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\	SUCCESS	FileBothDirectoryInformation
6586	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\	SUCCESS	
6587	17:16:25	setup.exe	IRP_MJ_CLOSE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\	SUCCESS	
6588	17:16:25	setup.exe	FASTIO_QUERY_STA...	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp	SUCCESS	Size: 789480
6589	17:16:25	setup.exe	FASTIO_QUERY_STA...	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp	SUCCESS	Size: 789480
6590	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp.Manifest	FILE NOT...	Attributes: Any Options: Open
6591	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp.Manifest	FILE NOT...	Attributes: Any Options: Open
6592	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp.Config	FILE NOT...	Attributes: Any Options: Open
6593	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp.Config	FILE NOT...	Attributes: Any Options: Open
6594	17:16:25	setup.exe	IRP_MJ_CREATE	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp.Config	FILE NOT...	Attributes: Any Options: Open
6595	17:16:25	setup.exe	IRP_MJ_CLEANUP	C:\DOCUME~1\assa\LOCALS~1\Temp\is-AEUKT.tmp\setup.tmp	SUCCESS	


7. Знаходимо дії над ліцензійним ключем програми (файлом Hidekey.dat) при виконанні процесу Setup.exe.


## II. Проведемо моніторинг інсталяції програми за допомогою Regmon.


Для цього виконаємо наступні дії:

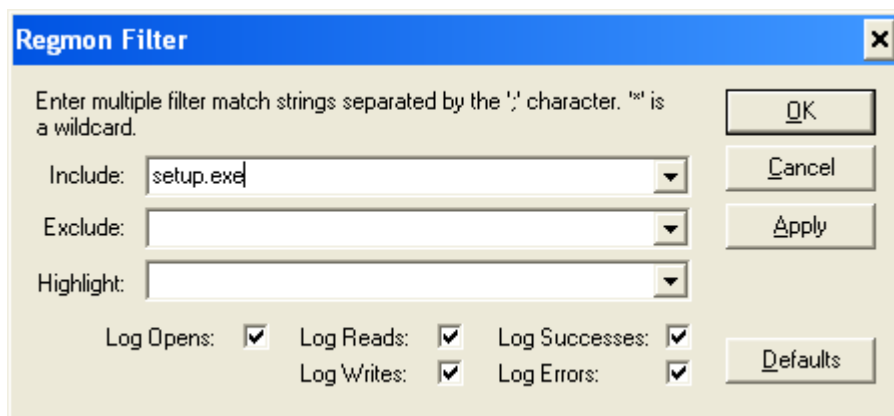
### 1. Запустимо Regmon.

Registry Monitor - Sysinternals: www.sysinternals.com						
File Edit Options Help						
#	Time	Process	Request	Path	Result	Other
1178	38.31560033	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\DhcpServer	SUCCESS	"10.139.0.1"
1179	38.31560835	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\DhcpServer	SUCCESS	"10.139.0.1"
1180	38.31561683	explorer.exe:1944	CloseKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}	SUCCESS	
1181	38.31565969	explorer.exe:1944	OpenKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}	SUCCESS	Access: 0x20...
1182	38.31566777	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}\EnableDHCP	SUCCESS	
1183	38.31567559	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}\DhcpServer	NOTFOUND	
1184	38.31568315	explorer.exe:1944	CloseKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}	SUCCESS	
1185	38.31572654	explorer.exe:1944	OpenKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}	SUCCESS	Access: 0x1
1186	38.31573572	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\AddressType	SUCCESS	0x0
1187	38.31574349	explorer.exe:1944	CloseKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}	SUCCESS	
1188	38.31634404	explorer.exe:1944	QueryValue	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage\Bind	BUFOVRFLOW	
1189	38.31635485	explorer.exe:1944	QueryValue	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage\Bind	BUFOVRFLOW	
1190	38.31636386	explorer.exe:1944	QueryValue	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage\Bind	SUCCESS	"Device\{C...
1191	38.31716230	explorer.exe:1944	OpenKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}	SUCCESS	Access: 0x20...
1192	38.31717087	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\EnableDHCP	SUCCESS	0x1
1193	38.31717942	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\LeaseObtainedT...	SUCCESS	0x504344A6
1194	38.31718788	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\LeaseTerminates...	SUCCESS	0x50437CE6
1195	38.31719572	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\DhcpServer	SUCCESS	"10.139.0.1"
1196	38.31720356	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\DhcpServer	SUCCESS	"10.139.0.1"
1197	38.31721158	explorer.exe:1944	CloseKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}	SUCCESS	
1198	38.31725299	explorer.exe:1944	OpenKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}	SUCCESS	Access: 0x20...
1199	38.31726104	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}\EnableDHCP	SUCCESS	0x0
1200	38.31726874	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}\DhcpServer	NOTFOUND	
1201	38.31727621	explorer.exe:1944	CloseKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}	SUCCESS	
1202	38.31733464	explorer.exe:1944	QueryValue	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage\Bind	BUFOVRFLOW	
1203	38.31734544	explorer.exe:1944	QueryValue	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage\Bind	BUFOVRFLOW	
1204	38.31735527	explorer.exe:1944	CloseKey	HKLM\SYSTEM\ControlSet001\Services\Tcpip\Linkage\Bind	SUCCESS	"Device\{C...
1205	38.31754092	explorer.exe:1944	OpenKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}	SUCCESS	Access: 0x20...
1206	38.31754934	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\EnableDHCP	SUCCESS	0x1
1207	38.31755786	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\LeaseObtainedT...	SUCCESS	0x504344A6
1208	38.31756644	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\LeaseTerminates...	SUCCESS	0x50437CE6
1209	38.31757421	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\DhcpServer	SUCCESS	"10.139.0.1"
1210	38.31758201	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}\DhcpServer	SUCCESS	"10.139.0.1"
1211	38.31759002	explorer.exe:1944	CloseKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{A4CF5798-8A43-478C-9056-EA21CE88FD53}	SUCCESS	
1212	38.31763098	explorer.exe:1944	OpenKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}	SUCCESS	Access: 0x20...
1213	38.31763894	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}\EnableDHCP	SUCCESS	
1214	38.31764672	explorer.exe:1944	QueryValue	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}\DhcpServer	NOTFOUND	
1215	38.31765409	explorer.exe:1944	CloseKey	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{13FA9F32-451F-4A99-8960-D9E3C3CE86ED}	SUCCESS	
1216	38.48243591	winlogon.exe:816	OpenKey	HKCU	SUCCESS	Access: 0x80...
1217	38.48245323	winlogon.exe:816	OpenKey	HKCU\AppDataEvents\Schemes\Apps\Default\MenuPopup\Current	SUCCESS	Access: 0x80...
1218	38.48246279	winlogon.exe:816	QueryValue	HKCU\AppDataEvents\Schemes\Apps\Default\MenuPopup\Current(Default)	SUCCESS	"
1219	38.48247373	winlogon.exe:816	CloseKey	HKCU\AppDataEvents\Schemes\Apps\Default\MenuPopup\Current	SUCCESS	
1220	38.48248163	winlogon.exe:816	CloseKey	HKCU	SUCCESS	
1221	38.48250415	winlogon.exe:816	OpenKey	HKCU	SUCCESS	Access: 0x80...
1222	38.48251608	winlogon.exe:816	OpenKey	HKCU\AppDataEvents\Schemes\Apps\Default\MenuPopup\Current\Active	NOTFOUND	
1223	38.48252175	winlogon.exe:816	QueryValue	HKCU(Default)	NOTFOUND	
1224	38.48252815	winlogon.exe:816	CloseKey	HKCU	SUCCESS	
1225	38.48254626	winlogon.exe:816	OpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion	SUCCESS	Access: 0x80...
1226	38.48255625	winlogon.exe:816	CloseKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Software\Microsoft\Windows\CurrentVersion	NOTFOUND	
1227	38.48256534	winlogon.exe:816	QueryValue	HKLM\Software\Microsoft\Windows\CurrentVersion\MediaPath	SUCCESS	"C:\WINDO...
1228	38.48257259	winlogon.exe:816	CloseKey	HKLM\Software\Microsoft\Windows\CurrentVersion	SUCCESS	
1229	38.67425755	winlogon.exe:816	OpenKey	HKCU	SUCCESS	Access: 0x80...
1230	38.67427520	winlogon.exe:816	OpenKey	HKCU\AppDataEvents\Schemes\Apps\Default\MenuPopup\Current	SUCCESS	Access: 0x80...
1231	38.67428487	winlogon.exe:816	QueryValue	HKCU\AppDataEvents\Schemes\Apps\Default\MenuPopup\Current(Default)	SUCCESS	"
1232	38.67429792	winlogon.exe:816	CloseKey	HKCU\AppDataEvents\Schemes\Apps\Default\MenuPopup\Current	SUCCESS	
1233	38.67430591	winlogon.exe:816	CloseKey	HKCU	SUCCESS	
1234	38.67432816	winlogon.exe:816	OpenKey	HKCU	SUCCESS	Access: 0x80...
1235	38.67433829	winlogon.exe:816	OpenKey	HKCU\AppDataEvents\Schemes\Apps\Default\MenuPopup\Current\Active	NOTFOUND	
1236	38.67434400	winlogon.exe:816	QueryValue	HKCU(Default)	NOTFOUND	
1237	38.67435029	winlogon.exe:816	CloseKey	HKCU	SUCCESS	
1238	38.67436872	winlogon.exe:816	OpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion	SUCCESS	Access: 0x80...
1239	38.67437879	winlogon.exe:816	OpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Software\Microsoft\Windows\CurrentVersion	NOTFOUND	
1240	38.67438815	winlogon.exe:816	QueryValue	HKLM\Software\Microsoft\Windows\CurrentVersion\MediaPath	SUCCESS	"C:\WINDO...
1241	38.67439537	winlogon.exe:816	CloseKey	HKLM\Software\Microsoft\Windows\CurrentVersion	SUCCESS	


2. Натискаємо на  для того, щоб зробити захват поточного стану процесів підсистеми файлів.

3. Натискаємо на  для того, щоб очистити список процесів.

4. Натискаємо кнопку фільтру  де задаємо той процес, який ми хочемо відстежити. У нашому випадку відстежуємо програму інсталяції (Setup.exe), яка створювалася у ЛР № 9.



Натискаємо «Apply».

5. Натискаємо на  для того, щоб зробити захват поточного стану процесу Setup.exe

6. Запускаємо програму Setup.exe, й дивимося які дії відбуваються при її запуску, та під час її виконання.

Registry Monitor - Sysinternals: www.sysinternals.com						
#	Time	Process	Request	Path	Result	Other
130	21.07.238986	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\wave6	NOTFOUND	
131	21.07.239726	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\wave7	NOTFOUND	
132	21.07.240463	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\wave8	NOTFOUND	
133	21.07.241200	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\wave9	NOTFOUND	
134	21.07.242197	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi	SUCCESS	"wdmaud.dr"
135	21.07.243075	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi	SUCCESS	"wdmaud.dr"
136	21.07.243898	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi1	SUCCESS	"wdmaud.dr"
137	21.07.244715	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi1	SUCCESS	"wdmaud.dr"
138	21.07.245536	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi2	NOTFOUND	
139	21.07.246318	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi3	NOTFOUND	
140	21.07.247087	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi4	NOTFOUND	
141	21.07.247860	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi5	NOTFOUND	
142	21.07.248645	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi6	NOTFOUND	
143	21.07.249426	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi7	NOTFOUND	
144	21.07.250217	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi8	NOTFOUND	
145	21.07.250996	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\midi9	NOTFOUND	
146	21.07.251837	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux	SUCCESS	"wdmaud.dr"
147	21.07.253098	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux	SUCCESS	"wdmaud.dr"
148	21.07.253890	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux1	SUCCESS	"wdmaud.dr"
149	21.07.254692	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux1	SUCCESS	"wdmaud.dr"
150	21.07.255463	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux2	NOTFOUND	
151	21.07.256197	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux3	NOTFOUND	
152	21.07.256929	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux4	NOTFOUND	
153	21.07.257676	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux5	NOTFOUND	
154	21.07.258406	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux6	NOTFOUND	
155	21.07.259134	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux7	NOTFOUND	
156	21.07.259863	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux8	NOTFOUND	
157	21.07.260686	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\aux9	NOTFOUND	
158	21.07.261471	setup.exe:844	OpenKey	HKLM\System\CurrentControlSet\Control\MediaProperties\PrivateProperties\Joystick\Wmm	SUCCESS	Access: 0xF003F
159	21.07.262400	setup.exe:844	OpenKey	HKLM\System\CurrentControlSet\Control\MediaProperties\PrivateProperties\Joystick\Wmm\wheel	SUCCESS	0x1
160	21.07.263645	setup.exe:844	CloseKey	HKLM\System\CurrentControlSet\Control\MediaProperties\PrivateProperties\Joystick\Wmm	SUCCESS	
161	21.07.264416	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer	SUCCESS	"wdmaud.dr"
162	21.07.265133	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer	SUCCESS	"wdmaud.dr"
163	21.07.265916	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer1	SUCCESS	"wdmaud.dr"
164	21.07.266692	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer1	SUCCESS	"wdmaud.dr"
165	21.07.267443	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer2	NOTFOUND	
166	21.07.268194	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer3	NOTFOUND	
167	21.07.268915	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer4	NOTFOUND	
168	21.07.269631	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer5	NOTFOUND	
169	21.07.270378	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer6	NOTFOUND	
170	21.07.271149	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer7	NOTFOUND	
171	21.07.271911	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer8	NOTFOUND	
172	21.07.272659	setup.exe:844	Query/Value	HKLM\Software\Microsoft\Windows NT\CurrentVersion\DRIVERS32\mixer9	NOTFOUND	
173	21.07.273411	setup.exe:844	OpenKey	HKCU	SUCCESS	Access: 0x2000000
174	21.07.274160	setup.exe:844	CreateKey	HKCU\Software\Microsoft\Multimedia\Audio	SUCCESS	Access: 0x20007

7. Знаходимо дії над ліцензійним ключем програми (реєстр HKEY\_LOCAL\_MACHINE\SOFTWARE\KNTU\NAME\)) при виконанні процесу Setup.exe.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Скріншот реалізованих завдань у Filemon та Regmon.

Опис дій над файлом Hidekey.dat та реєстром  
HKEY\_LOCAL\_MACHINE\SOFTWARE\KNTU\NAME\

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованих поставлених завдань.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

### **КОНТРОЛЬНІ ПИТАННЯ**

1. Що таке моніторинг ОС?
2. Що таке Filemon, Regmon, Process Explorer?
3. Що таке Process Monitor?
4. Що таке система повідомлень ОС Windows?
5. Що таке файлова система Microsoft Windows?
6. Приведіть приклади й історію програмного забезпечення фірми Sysinternals?



## Лабораторна робота №12 (семестр 6)

**ТЕМА:** Використання файлів конфігурації ОС Windows.

**ЦІЛЬ:** Вивчення можливостей файлів конфігурації.

**ЗНАТИ:** Мову програмування C або Pascal відповідно до обраної RAD.  
Основи адміністрування ОС Windows.

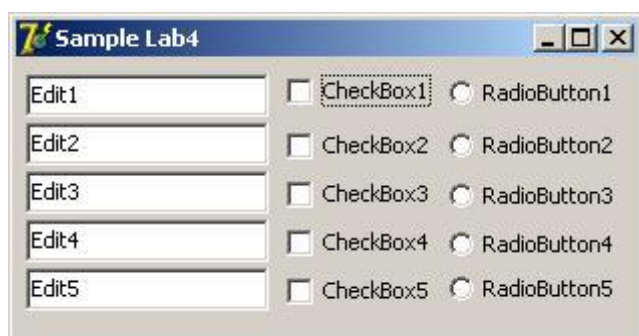
### ТЕОРЕТИЧНІ ВІДОМОСТІ:

1. WWW.GOOGLE.UA;
2. WWW.BING.COM;
3. UK.WIKIPEDIA.ORG;
4. <http://ru.wikipedia.org/wiki/.ini>

### ЗАВДАННЯ.

Специфікація програми взаємодії з файлом конфігурації.

На формі програми розташовані компоненти з довільним текстом (на вибір студента): Checkbox (5 шт.); Radiobutton (5 шт.); Edit (3 шт.).



Забезпечити збереження поточного статусу (Checkbox), положення (Radiobutton) і тексту (Edit) у файлі конфігурації програми Project.ini. Таким чином, щоб при закритті програми, та повторному відкритті ці результати

відображалися у програмі. Крім цього програма повинна зберігати поточну позицію щодо робочого стола ОС.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Скріншот реалізованих завдань та файлу конфігурації програми Project.ini.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованих поставлених завдань.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

**КОНТРОЛЬНІ ПИТАННЯ**

1. Що таке й навіщо необхідний файл конфігурації?
2. У яких випадках застосовують файли конфігурації?
3. Як файл конфігурації може пошкодити програму?
4. Приведіть приклади відомих форматів файлів конфігурації (у різних ОС)?
5. Який формат файлів конфігурації щонайкраще підійде для складної ієрархічної структури?
6. Що таке секції, параметри й значення у файлах конфігурації?

## **Лабораторна робота №13 (семестр 6)**

**ТЕМА:** Створення програми меню для носія інформації

**ЦІЛЬ:** Вивчення можливостей створення програм меню.

**ЗНАТИ:** Мову програмування C або Pascal відповідно до обраної RAD.

**ТЕОРЕТИЧНІ ВІДОМОСТІ:** Теоретичні відомості наведені у лекції № 8 (3 семестр)

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

1. WWW.GOOGLE.UA;
2. WWW.BING.COM;
3. UK.WIKIPEDIA.ORG;
4. <http://ru.wikipedia.org/wiki/Autorun.inf>

### **ЗАВДАННЯ.**

1. Створити файл autorun.inf для забезпечення автоматичного запуску програм з носіїв інформації (USB флеш накопичувач – розмовний термін “Флешка”). Установити параметри action, icon, Defaulticon, label, open, Useautoplay, shellexecute, shell файлу autorun.inf по необхідності.

2. Створити програму для носія інформації. Яка буде автоматично виконуватись при підключенні носія до ОС.

Функціональні можливості:

- запуск програми (на вибір) з носія інформації;
- запуск мультимедіа файлів;
- виклик провідника Windows з заданою директорією на носії.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Скріншот реалізованих завдань.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованих поставлених завдань.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче.

## **КОНТРОЛЬНІ ПИТАННЯ**

1. Як саме працює автоматичний запуск програм з носіїв інформації?
2. Як відключити автозапуск програм з носіїв інформації?
3. Де повинен перебувати файл Autorun.inf для автозапуску?
4. Формат файлу (структурні блоки) Autorun.inf?
5. Що таке параметр action, блоку autorun?
6. Що таке параметр icon, блоку autorun?
7. Що таке параметр Defaulticon, блоку autorun?
8. Що таке параметр label, блоку autorun?
9. Що таке параметр open, блоку autorun?
10. Що таке параметр Useautoplay, блоку autorun?
11. Що таке параметр shellexecute, блоку autorun?

## Лабораторна робота №14 (семестр 6)

**ТЕМА:** Створення власного формату файлу

**ЦІЛЬ:** Вивчення можливостей створення й зберігання власних типів даних.

**ЗНАТИ:** Мова програмування C або Pascal відповідно до обраної RAD.

### ТЕОРЕТИЧНІ ВІДОМОСТІ:

У зв'язку з великим обсягом інформації використовувати електронну документацію (погоджувати з лектором).

1. WWW.GOOGLE.UA;
2. WWW.BING.COM;
3. UK.WIKIPEDIA.ORG;
4. [http://ru.wikipedia.org/wiki/Тип\\_данных](http://ru.wikipedia.org/wiki/Тип_данных)
5. [http://ru.wikipedia.org/wiki/Запись\\_\(тип\\_данных\)](http://ru.wikipedia.org/wiki/Запись_(тип_данных))

### ЗАВДАННЯ.

1. Створити власний формат файлу для зберігання різних даних (свою базу даних), використовуючи структури обраного мови програмування.

Використання баз даних заборонене! У тому числі й вбудованих у мови програмування.

Формат файлу видається викладачем індивідуально.

2. Створити програму, яка зможе обробляти створений формат.

Приклад завдання.

Створити формат файлу «Книга контактів» для збереження наступної інформації:

- Index: Ціле число ( від 1...N), номер запису контакту;
- FIO: Текстове поле (15 символів), ПІБ контакту;
- Home phone 1: Текстове поле (10 символів), домашня адреса 1;
- Home phone 2: Текстове поле (10 символів), домашня адреса 2;

- Business phone 1: Текстове поле (10 символів), діловий телефон 1;
- Business phone 2: Текстове поле (10 символів), діловий телефон 2;
- Home adress: Текстове поле (100 символів), домашня адреса;
- Notes: Текстове поле (Динамічне, необмежене), замітки. Розширення файлу \*.nts.

**Звіт з лабораторної роботи повинен містити наступні елементи**

Оформлена титульна сторінка.

Завдання.

Назва проекту, короткий його опис.

Скріншот реалізованих завдань.

**Лабораторна робота вважається зарахованою при виконанні наступних умов:**

Написана на позитивну оцінку летуча контрольна робота.

Наявність звіту, оформленого згідно наведених вище вимог.

Наявність реалізованих поставлених завдань.

Співбесіда з викладачем, який приймає лабораторну роботу на наявність знань з даної тематики. (Або відповідь на контрольні питання наведені нижче).

### **КОНТРОЛЬНІ ПИТАННЯ**

1. Що таке тип даних?
2. Що таке користувацький(власний) тип даних?
3. Приведіть класифікацію типів даних?
4. Що таке динамічна ідентифікація типу даних?
5. Що таке статична типізація?
6. Що таке динамічна типізація?
7. Що таке сурова типізація?
8. Що таке слабка типізація?
9. Що таке приведення типів даних?
10. Бувають мови без типів даних?

### Список використаної літератури

1. Иан Соммервилл. Инженерия программного обеспечения, 2002.
2. Орлов С. А. Технологии разработки программного обеспечения: Разработка сложных программных систем Изд. 3-е, 2004.
3. Эрик Дж. Брауде. Технология разработки программного обеспечения, 2004.
4. Липаев, В.В. Программная инженерия. Методологические основы : Учеб. / В. В. Липаев ; Гос. ун-т – Высшая школа экономики. – М. : ТЕИС, 2006. – 608 с. – 1000 экз.
5. Гленфорд Майерс, Том Баджетт, Кори Сандлер Искусство тестирования программ, 3-е издание = The Art of Software Testing, 3rd Edition. – М.: «Диалектика», 2012. – 272 с. – ISBN 978-5-8459-1796-6.
6. Лайза Криспин, Джанет Грегори Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = Agile Testing: A Practical Guide for Testers and Agile Teams. – М.: «Вильямс», 2010. – 464 с. – (Addison-Wesley Signature Series). – 1000 экз. – ISBN 978-5-8459-1625-9.
7. Канер Кем, Фолк Джек, Нгуен Енг Кек Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. – Киев: ДиаСофт, 2001. – 544 с. – ISBN 9667393879.
8. Калбертсон Роберт, Браун Крис, Кобб Гэри Быстрое тестирование. – М.: «Вильямс», 2002. – 374 с. – ISBN 5-8459-0336-X.
9. Синицын С. В., Налютин Н. Ю. Верификация программного обеспечения. – М.: БИНОМ, 2008. – 368 с. – ISBN 978-5-94774-825-3.
10. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. – СПб.: Питер, 2004. – 320 с. – ISBN 5-94723-698-2.
11. Братищенко В.В. Проектирование информационных систем. – Иркутск: Изд-во БГУЭП, 2004. – 84 с.

12. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2000.
13. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. – М.: Интернет-университет информационных технологий – ИНТУИТ.ру, 2005.
14. Мишенин А.И. Теория экономических информационных систем. – М.: Финансы и статистика, 2000. – 240 с.
15. Мирошниченко Е. А. Технологии программирования: учебное пособие / Е. А. Мирошниченко. – 2-е изд., испр. и доп. – Томск: Изд-во Томского политехнического университета, 2008. – 128 с.
16. Ларман К. Итеративная и инкрементальная разработка: краткая история / К. Ларман, В. Базили // Открытые системы. – 2003.— N 9.
17. Орлик С. Введение в программную инженерию и управление жизненным циклом ПО.
18. Государственный стандарт РФ ГОСТ Р ИСО\_МЭК 12207-99 'Информационная технология. Процессы жизненного цикла программных средств' 1999.
19. Крачтен Ф., Оббинк Х., Стаффорд Д. Ретроспектива программных архитектур.
20. Мартин Фаулер Шаблоны корпоративных приложений = Patterns of Enterprise Application Architecture (Addison-Wesley Signature Series). – М.: «Вильямс», 2009. – С. 544. – ISBN 0-321-12742-0.
21. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес Приемы объектно-ориентированного проектирования. Паттерны проектирования = Design Patterns: Elements of Reusable Object-Oriented Software. – СПб: «Питер», 2007. – С. 366. – ISBN 978-5-469-01136-1 (также ISBN 5-272-00355-1).
22. Марк Гранд Шаблоны проектирования в JAVA. Каталог популярных шаблонов проектирования, проиллюстрированных при помощи UML = Patterns in Java, Volume 1. A Catalog of Reusable Design Patterns



Illustrated with UML. – М.: «Новое знание», 2004. – С. 560. – ISBN 5-94735-047-5.

23. Крэг Ларман Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. – М.: «Вильямс», 2006. – С. 736. – ISBN 0-13-148906-2.

24. Джошуа Кериевски Рефакторинг с использованием шаблонов (паттернов проектирования) = Refactoring to Patterns (Addison-Wesley Signature Series). – М.: «Вильямс», 2006. – С. 400. —ISBN 0-321-21335-1.

25. Скотт В. Эмблер, Прамодкумар Дж. Садаладж Рефакторинг баз данных: эволюционное проектирование = Refactoring Databases: Evolutionary Database Design (Addison-Wesley Signature Series). —М.: «Вильямс», 2007. – С. 368. – ISBN 0-321-29353-3.

26. Тереза Нейл, Билл Скотт Проектирование веб-интерфейсов = Designing Web Interfaces. – М.: Символ-плюс, 2010. – С. 352. —ISBN 978-5-93286-172-1

27. Степанов А.Н. Информатика: Учебник для вузов. 4-е изд. – СПб.: Питер, 2006. – 684 с.: ил. ISBN 5-94723-898-5.

28. ГОСТ 19.001-77. ЕСПД. Общие положения.

29. ГОСТ 19.003-80. ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические.

30. ГОСТ 19.005-85. ЕСПД. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения.

31. ГОСТ 19.101-77. ЕСПД. Виды программ и программных документов.

32. ГОСТ 19.102-77. ЕСПД. Стадии разработки.

33. ГОСТ 19.103-77. ЕСПД. Обозначение программ и программных документов.

34. ГОСТ 19.104-78. ЕСПД. Основные надписи.

35. ГОСТ 19.105-78. ЕСПД. Общие требования к программным документам.
36. ГОСТ 19.106-78. ЕСПД. Требования к программным документам, выполненным печатным способом.
37. ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению.
38. ГОСТ 19.202-78. ЕСПД. Спецификация. Требования к содержанию и оформлению.
39. ГОСТ 19.301-79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению.
40. ГОСТ 19.401-78. ЕСПД. Текст программы. Требования к содержанию и оформлению.
41. ГОСТ 19.402-78. ЕСПД. Описание программы.
42. ГОСТ 19.403-79. ЕСПД. Ведомость держателей подлинников.
43. ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
44. ГОСТ 19.501-78. ЕСПД. Формуляр. Требования к содержанию и оформлению.
45. ГОСТ 19.502-78. ЕСПД. Описание применения. Требования к содержанию и оформлению.
46. ГОСТ 19.503-79. ЕСПД. Руководство системного программиста. Требования к содержанию и оформлению.
47. ГОСТ 19.504-79. ЕСПД. Руководство программиста. Требования к содержанию и оформлению.
48. ГОСТ 19.505-79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению.
49. ГОСТ 19.506-79. ЕСПД. Описание языка. Требования к содержанию и оформлению.
50. ГОСТ 19.507-79. ЕСПД. Ведомость эксплуатационных документов.

51. ГОСТ 19.508-79. ЕСПД. Руководство по техническому обслуживанию. Требования к содержанию и оформлению.
52. ГОСТ 19.601-78. ЕСПД. Общие правила дублирования, учета и хранения.
53. ГОСТ 19.602-78. ЕСПД. Правила дублирования, учета и хранения программных документов, выполненных печатным способом.
54. ГОСТ 19.603-78. ЕСПД. Общие правила внесения изменений.
55. ГОСТ 19.604-78. ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом.
56. ГОСТ 19.701-90 (ИСО 5807-85). ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.